91602
new 82-10

LEGAL NOTICE

This report was prepared by Ontario Hydro and, while effort has been made to assure the accuracy and applicability of the material contained herein, neither Ontario Hydro nor any person acting on its behalf:

(a) makes any warranty, express or implied with respect to the use of any information, apparatus, method or process disclosed in this report or that such use may not infringe privately owned rights; or

(b) assumes any liabilities with respect to the use of, or for damages resulting from the use of, any information, apparatus, method or process disclosed in this report.

ontario hydro
research division

COMPLIMENTARY COPY
Please Credit Ontario Hydro Research if Reproduced in Whole or in Part

To: Mr. M.L. Walsh
Manager, Technical Safety Standards Department
Health and Safety Division

A DATA ACQUISITION SYSTEM FOR RESIDENTIAL MAGNETIC FIELD SOURCE CHARACTERIZATION

K.E. Donnelly

A Data Acquisition System has been designed and built for a project to characterize the sources of residential magnetic fields. The instrument is portable computer based, and has three current monitoring channels, a 3-axis magnetic field probe, and an auxiliary magnetic field coil to monitor fields from inaccessible currents. Data on harmonics and phases are obtained. It is capable of recording unattended for extended periods of time.

PURPOSE OF INSTRUMENT

Ontario Hydro's Electric and Magnetic Field Risk Assessment Program (EMFRAP) includes an investigation of the sources of magnetic fields in Ontario residences. This Source Characterization Program will establish the factors which determine the level of magnetic field exposure and their prevalence, contribute to the development of improved surrogate measures for epidemiological investigations, and suggest methods of reducing high values of field exposure, if required, by changes in grounding and distribution practices.

To obtain magnetic field and current data for this investigation, a Data Acquisition System has been designed and constructed. The objective was to produce an instrument which would monitor and record several channels of 60 Hz current signals as the prime sources of fields in a subject house, along with magnetic field at a representative location for corroborative purposes, and other channels as required to monitor fields originating outside the house.

job	file	date	report no.
16-229	817.22	October 24, 1989	89-173-K

The present instrument is primarily designed for use in residences, but is to include the flexibility to be reconfigured for use in other environments such as schools, industries or other commercial buildings.

DESIGN CONFIGURATION

The instrument is personal computer based to make use of commercially available multichannel signal acquisition boards, permit flexible signal processing algorithms to be developed, and provide storage for unattended data logging.

Signals are to be analysed for harmonic content as well as 60 Hz magnitude, and phase information is also desired to aid in the identification of sources. Information regarding the polarization of magnetic fields is also desirable in three-phase environments or where significant power factor corrections occur.

Signal Channels

The signal channels in the instrument are as follows:

1. Voltage Reference. This signal is derived from the 110 V household power supply in order to provide a local reference for phase and harmonic content (voltage). The phase of the 60 Hz Fourier component is used as the reference.

2. Current Probe 1. A 1000:1 clamp-on current transformer is used as current transducer. An operational amplifier configured for very low input impedance is used in the input stage for best ratio and phase accuracy with a variety of commercially available CT's. The amplifier is configured to give a flat frequency response from 60-600 Hz.

3. Current Probe 2. The same as Current Probe 1.

4. Current Probe 3. The same as Current Probe 1.

5. Magnetic Field Coil X. A 2000 turn coil of Number 34 magnet wire has been wound on a 10 centimetre square form, and shielded against electrostatic pickup. The amplifier input impedance and frequency shaping components are chosen to give a flat response for the coil-amplifier combination from 60-600 Hz.

6. Magnetic Field Coil Y. The same as Field Coil X.

7. Magnetic Field Coil Z. The same as Field Coil X. Coils X, Y and Z are mounted together in an orthogonal configuration.

8. Auxiliary Field Coil. Also the same as Field Coil X. This input is used for monitoring currents, such as distribution line net currents, which cannot easily be measured directly. This will be positioned outside the house, or remote from known current paths such as grounding conductors or water pipes.

9. Total Magnetic Field. This is not a real signal channel, but consists of the total field computed from Field Coils X, Y and Z. It is treated as separate data channel for display purposes.

Hardware Implementation

A Data Translation 2825 Analog-Digital interface board was chosen for signal acquisition. This board has high data throughput (40 KHz overall), 8 differential or 16 single-ended inputs, 12 bit digitization and programmable gain settings of 1, 10, 100 or 500. Direct Memory Access and an on-board clock are additional features.

The portable computer chosen is a SHARP model PC-7220; reasons for selection are as follows: a 16-bit AT expansion slot, required by the DT-2825 board; a small footprint for field use; a keyboard which may be closed during unattended operation; and a rugged design for reliability in field use.

Signal conditioning amplifiers and phase reference source are contained in a separate electronics package. These are discussed in Appendix 2.

A schematic of the hardware configuration is shown in Figure 1. Eight channels are presently implemented; this can be extended to sixteen by adding additional signal conditioning amplifiers and expanding various display menus.

PERFORMANCE SPECIFICATIONS

Frequency Response

Current amplifiers are designed for a response which is flat within 1 % in ratio from 60 Hz to 600 Hz. Phase accuracy of ± 2 degrees at 60 Hz and ± 10 degrees up to 300 Hz is sufficient for this instrument's purposes. The same specifications apply to the magnetic field coil and amplifier combination together; the frequency factor in coil emf is compensated by appropriate shaping in amplifier response to give a flat characteristic through the range of interest.

Present program parameters give a sampling rate of 960 Hz on each channel; this corresponds to a Nyquist frequency of 480 Hz, and a usable frequency limit of approximately 380 Hz. Higher sampling rates are easily programmed, however, Fast Fourier Transforms require a doubling of sampling rate and consequent doubling of storage requirements. This can be incorporated at a later time if required.

Current and Field Ranges

The 12-bit digitization of the A/D sampling board determines the basic dynamic range of the instrument; this corresponds to a 2048 times range from peak to digitization limit for a bipolar signal.

A maximum current signal of 50 Amps has been selected for the prototype instrument. The corresponding digitization limit is 25 mAmp; the minimum current measurable with moderate accuracy (slightly better than 10%, because of averaging) is then approximately 0.25 Amp. This range is certainly adequate for residential studies, but may need modification in other electrical environments.

The magnetic field range which is expected poses somewhat stricter constraints on the instrument range. An upper limit of 10 microTesla has been chosen, with a corresponding digitization limit of 0.005 microTesla. The minimum field measurable with moderate accuracy is then approximately 0.05 microTesla. Lower fields will likely be encountered, and may be measured with decreasing accuracy down to the digitization limit.

Sampling Interval

All eight channels are sampled virtually simultaneously for two cycles, the waveforms processed by a FFT algorithm, and the results displayed in real-time. Details of display options are given below. The process limiting the maximum acquisition rate is the graphics routine, giving a minimum interval between sampled events of slightly under ten seconds. Maximum interval between samples is once per day, due to clock algorithms.

It should be borne in mind that an array in computer RAM is filled and periodically dumped to disk; this process determines the amount of data exposed to the risk of power failure. At present, the maximum array (64 KBytes) is used to give the largest single time frames for later viewing. This stores 128 sampling events of eight channels and eight harmonics, magnitude and phase at four bytes per real number.

Logging Duration

Duration of a data-logging run is set by computer hard-disk storage capacity. As outlined above, for the present channel and harmonic configuration, disk usage is 64 KBytes per 128 events (excluding minor file overhead requirements), or 2048 events per MByte. This translates to 1.4 days per MegaByte at one minute sampling intervals. Approximately 11 days of data could be acquired on the present 20 Mbyte hard disk at this rate, if properly partitioned and restricted only to the operating program. No provision has yet been made for long-term storage beyond floppy disk capacity.

PROGRAM DESIGN

The program was coded in ASYST Version 2.01, a high-level language permitting simple A/D board control, direct memory access, file creation and read/write access. Many analytic processes occur in this package as simply callable routines. Simple graphics routines allow rapid prototyping of displays.

Sampling occurs virtually simultaneously on the eight channels previously described; an interleaved multiplex scheme is used. Each waveform is sampled for two cycles and processed by a Fast Fourier Transform Algorithm. Fourier magnitudes and phases are stored as the quantities of interest. Phases are adjusted for sampling delays between channels. Conversion factors are incorporated in the program for translation from the signal domain to the units of interest, using the nominally assigned signal ranges. Calibration is done with adjustments on the signal conditioning circuits.

Program functions are illustrated in Figure 2. The instrument has three basic modes of operation:

1. Real-time display. This function displays either waveform or spectrum and phase for any selected channel. The display is updated at intervals of three seconds, or may be held. The function is intended for equipment setting up and testing, and for exploratory use in a residence. A sample display is shown in Figure 3.

2. Data Logging. This function allows the creation of a data storage file and input of identification information. The interval between samples, and the run duration are also set by the operator at this stage. Once data acquisition is started, sample events are accumulated in an array in RAM, as outlined above (see PERFORMANCE SPECIFICATIONS - Sampling Interval); 128 events are stored, then the

entire array is dumped to a disk file. A date and time stamp are written along with the array. Further arrays are accumulated and added to the file, along with their respective date and time stamps. Each array constitutes a frame which may be accessed for later viewing.

During data acquisition, the data traces for the current frame may be viewed. Any two channels may be selected, however,only the magnitude of the 60 Hz component is available. This choice avoids the complexities of a multi-level menu operating simultaneously with a synchronized data acquisition loop. The display is similar to that in Figure 4.

3. Data Display. This function allows viewing of previously stored data. The file may be scanned through, frame by frame. Any two channels may be displayed simultaneously, and magnitude or phase may be displayed for fundamental or any harmonic. Two examples, showing magnitude and phase during the same time frame for a current and a magnetic field channel, are shown in Figures 4 and 5.

A description in more detail and listing of the program are given in the Appendices.

Approved:

A.F. Baljet
Manager
Electrical Research Department

Submitted:

K.E. Donnelly
Physicist
Science Section
Electrical Research Department

KED:td

APPENDIX 1

Software Description

A description of the Data Acquisition System's operating program will be given in this appendix. This is intended to give an overview of the program operation for general reference, and sufficient detail for an experienced ASYST programmer to make enhancements or modifications without excessive orientation time.

As the program is driven from menus which are self-explanatory to its intended engineering user, detailed descriptions of all functions will not be given. A short start-up procedure will be given outlining how to get the program into operation.

Further, as this is a prototype instrument, and intended for use by an engineering team for investigative purposes, a number of enhancements and finishing touches which might be expected from a general commercial instrument have not been incorporated.

The program was written in ASYST Version 2.01, which incorporates extensive enhancements over Version 1. software.

Program Overview

The instrument's three main functions, accessed from the main menu, were shown in Figure 2. These are Real-time Display of signal waveforms or their harmonic components, Data Logging to sample at preset intervals and record for an extended period, and Data Display to recall and display previously recorded data. Data Logging mode also provides an ongoing display of signal levels from the current logging run and may be used to observe fluctuations or short-term trends.

Figures A1 to A3 are program flow charts which show further menu choices within each of these operating modes, along with processes which automatically execute between operator interaction.

Real-time Display begins with selection of a channel for viewing, followed by choice of a display type and execution of the sampling process. The display either updates at three second intervals or halts. The operator follows menu choices to change the channel or display type.

Data Logging begins by creating a storage file on hard disk. The operator inputs a number of file identifiers and two parameters which control the acquisition frequency and run duration. The interactive dialogue is shown in Figure A4. During acquisition, an array in computer RAM is filled with 128 events (determined by the maximum ASYST array size, the number of channels recorded and the number of harmonics) separated by the acquisition interval and copied to disk file; this sequence of 128 events is designated a data frame. The process repeats up to the specified run duration. The number 128 is determined by the maximum ASYST array size, the number of channels recorded and the number of harmonics, and the number of events will take on a different value if these parameters are changed within the program.

During Data Logging, new channels may be chosen for display. Also available to the operator are the options of exiting, with or without a save of the current data frame.

In Data Display mode, the operator specifies a file for viewing. After opening the file and reading the identifying comments and various file parameters, the program displays the first data frame for a set of default channels and display types. The display is now menu driven, with options of scanning through data frame by frame, changing channels, or changing the type of data (magnitude or phase, harmonic number) displayed.

Data Flow Diagram

A data flow diagram covering the main analytic processes is given in Figure A5. This is given as a programmer's reference to trace the transformations of data through different variable names and different processing or calibration steps.

Analytic Processes

Sampling and the main analytic processes occur in the programmed ASYST word (or subroutine) "SAMPLE", which in turn calls other words to carry out a number of individual data processing tasks. The word SAMPLE is listed in Figure A6; only a few words need explanation beyond the program annotations.

The multiplexed sampling process samples each channel in turn at equal time intervals before returning to the first channel for another round of samples. A sufficient number of rounds are completed within a few cycles of the 60 Hz waveforms. By interleaving the samples from different channels in this way, nearly simultaneous samples are taken and simple adjustments allow accurate phase comparisons to be made between channels.

The word RELATIVE.PHASE references all phases to the voltage reference spectrum. Since this may have harmonic content, the 60 Hz phase provides a more accurate reference than a zero-crossing. PHASE.SHIFT adjusts the phases of all signal channels for finite time intervals between the multiplexed samples. PHASE.ZERO sets the phases of small magnitude signals equal to zero rather than assign a phase of questionable accuracy. Finally, PHASE.RANGE returns all phases to the range (-180, 180) degrees if earlier manipulations have shifted them out.

Program Modules

The data acquisition program consists of four modules which are kept separate for ease of editing and debugging. These are

SAMPLA: which contains the sampling and harmonic analysis routines,

SAMPLB: which contains programming for the Main Menu and for Real-time Display,

SAMPLC: which contains Data Logging routines, and

SAMPLD: which contains Data Display.

The program is menu controlled, hence the modules are designed to pass control back and forth. However, in testing and debugging, it is simple to enter at the appropriate menu word for each module.

Operating Instructions

A brief set of start-up instructions are given for the present configuration:

1. From the directory C:\ASYST201\, enter the command ASYSTBIG which accesses an ASYST system configured with the overlays and memory allocations required for the data acquisition program. The ASYST security block must be in place on the Parallel Printer Output connector.

2. To load the program after receiving the OK prompt, enter LOAD LOADR.PRG. This executes a small program which loads the four program modules SAMPLA, SAMPLB, SAMPLC, SAMPLD.

3. Enter GO to begin program execution. This command may occasionally be needed to restart the program after a long series of menu choices in real-time display mode.

Additional hints:

1. When creating a data file for logging purposes, a path may be specified such as D:\LOGGING\filename to use a disk with sufficient storage. This becomes the default path for the session. When retrieving a file for display at a later date, the path will again be needed. Initially, C:\ASYST201\ will be the default directory.

2. Before leaving the system for unattended operation, turn down screen display Brightness and Contrast to extend the display life. The SHARP PC-7200 may be left with the keyboard closed to discourage tampering.

APPENDIX 2[1]

HARDWARE

To transform the signals from the current transformers and magnetic field antennas to voltage levels compatible with the sampling A/D board an external signal conditioning box was built. In this Signal Conditioning Box there are 3 basic circuits, which are the:

Current Transformer Amplifier
Voltage Reference Amplifier
Magnetic Field Pickup Amplifier

These circuits and their performance will be described below.

Current Transformer Amplifier

This circuit is a trans-impedance amplifier designed to present near zero impedance to the current transformer, this optimizes ratio and phase accuracy. In this box there are three of these amplifiers allowing three current transformers to be connected to the computer at the same time.

To test the frequency response of the circuit a AEMC Model #100.185 current transformer was connected to the input of the current transformer amplifier. This current transformer measured the current in a resistor network driven by a HP35660A Spectrum Analyzer and a Yamaha PC5002M Power Amplifier. The current in the resistor network was also measured with a Pearson 411 Current Transformer, which was used as a reference signal. The output of the signal conditioning box was compared to the output of the Pearson current transformer. The frequency response of this circuit can be seen in Figure A7, and Figure A8 for 2 different currents.

By adjusting the potentiometer an output of 0.141 V/A was achieved. Further adjustment can be used to provide overall system calibration.

Magnetic Field Pickup Amplifiers

This circuit is a high impedance input device to allow the measurement of the open circuit voltage of a magnetic field pickup. This pickup was a 2000 turn coil on a 10 cm. square frame. In this box there are 5 of these circuits, however, only 4 of these are presently being used. This allows the measurement of fields along the X, Y, and Z axis as well as an auxiliary coil.

To test the frequency response of the circuit and magnetic field pickups a Helmholtz coil was driven by a HP35660A Spectrum Analyzer and a Yamaha PC5002M Power Amplifier. The magnetic field produced by the Helmholtz coil was mathematically determined to be 20.6μT/A. To measure the current in the coil a Pearson Model 411 Current Transformer was used. The spectrum analyzer was connected in such a way as to compare the output of the magnetic field pickup with the output of the current transformer. Since the magnetic field pickup coils response is dependent on the construction all four coils were tested for their response and the frequency responses shown in Figures A9, and A10. The standard deviation between the curves is also shown in the appropriate Figure.

---

1. This appendix was written by R.R. Scheer.

By adjusting the gain of the amplifier an output of 0.0707 volts/T was achieved. Again further adjustment can be used to provide overall system calibration.

Voltage Reference Amplifier

This amplifier is connected directly to the line voltage through a resistor divider network. It is designed to provide an output of 5 V with no phase shift. In this way the phase of the currents and fields can be measured with respect to household power supply. This measurement may vary by 180^0 depending on which phase of the supply the signal conditioning box is plugged into.

The circuit diagrams for these circuits is shown in Figure A11. The power, ground and bypass capacitors have been omitted for clarity.

FILE SAMPLA

```
\ MENU-DRIVEN WITH REAL-TIME AND SCROLLED GRAPHICS
\ PROGRAM TO SAMPLE MULTIPLE CHANNELS, DO HARMONIC ANALYSIS AND
\ REPORT AMPLITUDE AND PHASE

\ DMA VERSION, X 10 GAIN IN COIL CHANNELS

\ IMPROVED MENUS AND GRAPHICS, PHYSICAL UNITS INCORPORATED

\ AUTHOR K.E. DONNELLY FEBRUARY 28, 1989


\
\ set up channel list
integer scalar CHAN.BEGIN SCALAR CHAN.END SCALAR NO.CHANNELS
0 CHAN.BEGIN :=                    \ A/D channel 0 is used
7 CHAN.END :=                      \ channel 7 is last
CHAN.END CHAN.BEGIN - 1 + NO.CHANNELS := \ calc. no. of channels

\ set up sampling window, sampling rate for multiple channel acquisition
\ samples are taken continuously on interleaved channels for NCYC cycles

INTEGER SCALAR NCYC
2 NCYC :=                          \ sample for NCYC cycles
\
INTEGER SCALAR NPERCYC
16 NPERCYC :=
\
INTEGER SCALAR SAMPLES.PER.CHANNEL  \ # of samples per shot (per channel)
INTEGER SCALAR NSAMPLES             \ # of samples per shot (total, all channels)
INTEGER SCALAR NO.HARMONICS         \ # of harmonics from Fourier Analysis
NCYC NPERCYC *  SAMPLES.PER.CHANNEL :=
SAMPLES.PER.CHANNEL  NO.CHANNELS *   NSAMPLES :=
NPERCYC 2 / NO.HARMONICS :=         \ # of frequencies = 1/2 # of sample points
\
REAL SCALAR SRATE                   \ sampling rate (multiplexed total)
60. NPERCYC * NO.CHANNELS *   SRATE :=
\
REAL SCALAR SAMPL.DELAY             \ sampling delay (msec)
1000. SRATE /    SAMPL.DELAY :=
\
\ set up acquisition and storage arrays


INTEGER DIM[ SAMPLES.PER.CHANNEL , NO.CHANNELS ] DMA.ARRAY DATA.ARRAY
DP.REAL DIM[ SAMPLES.PER.CHANNEL , NO.CHANNELS ] ARRAY VOLT.ARRAY
DP.REAL DIM[ SAMPLES.PER.CHANNEL , NO.CHANNELS ] ARRAY SIGNAL.ARRAY
DP.REAL DIM[ NO.CHANNELS ] ARRAY SENSOR.CALIBN
DP.COMPLEX DIM[ NO.HARMONICS , NO.CHANNELS ] ARRAY FOURIER.COEFF
DP.REAL DIM[ NO.HARMONICS , NO.CHANNELS ] ARRAY FOURIER.MAG
DP.REAL DIM[ NO.HARMONICS , NO.CHANNELS ] ARRAY FOURIER.PHASE
DP.REAL DIM[ NO.HARMONICS , NO.CHANNELS ] ARRAY SIGNAL.MAG
DP.REAL DIM[ NO.HARMONICS ] ARRAY 3D.HARMONIC.MAGNITUDES

DP.REAL DIM[ SAMPLES.PER.CHANNEL ] ARRAY HANN.WINDOW.ARRAY


\ enter calibration factors for current and field sensors
\    Order: volts.ref, 3x current, 3x field coils, auxiliary coil
\    Defined as  current (Amps rms) = K1 x volts(rms) at ADC input
\                field (uT rms)     = K2 x volts(rms)

   SENSOR.CALIBN @[ 1 ]
   ENTER[ 1.0 , 7.07 , 7.07 , 7.07 , 1.414 , 1.414 , 1.414 , 1.414 ]
   DROP


\ set up magnitude criterion for setting phase = 0
REAL SCALAR NOISE.BASE
0.0001  NOISE.BASE :=              \ arbitrary, select by experience


\ 60 Hs component of voltage used as phase reference
REAL SCALAR PHASE.REF


\ create templates and initialize

DT2820                             \ declare A/D board series
```

```
integer dim[ 16. ] array GAINS&CHNLS  \ !!!!!!!!!!!!!!!!!!

CHAN.BEGIN CHAN.END A/D.TEMPLATE IN.BOARD    \ template named IN.BOARD

GAINS&CHNLS A/D.GAIN      \ !!!!!!!!!!!!!!!!!!!!
        DT2820.ABS.GAIN&CHNL    \ !!!!!!!!!!!!!!
A/D.INIT                  \ !!!!!!!!!!!!!!!!

: MAKE.INTO.GAIN/CHNL     \ !!!!!!!!!
   SWAP 16. * +           \ !!!!!!!!!
;                         \ !!!!!!!!!


GAINS&CHNLS @[ 1. ]               \ !!!!!!!!!!!!!!!!
ENTER[
     0 0 MAKE.INTO.GAIN/CHNL ,
     0 1 MAKE.INTO.GAIN/CHNL ,
     0 2 MAKE.INTO.GAIN/CHNL ,
     0 3 MAKE.INTO.GAIN/CHNL ,
     1 4 MAKE.INTO.GAIN/CHNL ,
     1 5 MAKE.INTO.GAIN/CHNL ,
     1 6 MAKE.INTO.GAIN/CHNL ,
     1 7 MAKE.INTO.GAIN/CHNL
   ]
DROP                      \ !!!!!!!!!!!!!!!!

: INIT
    IN.BOARD                        \ template name
    DATA.ARRAY DMA.TEMPLATE.BUFFER
    INT.CLOCK
    SAMPL.DELAY CONVERSION.DELAY          \ set conversion.delay
    A/D.INIT
;



: DT2825.ACQUIRE
 0 DATA.ARRAY :=
\  DATA.ARRAY .
\  BREAK

    INIT
    DAS.INIT
    IN.BOARD



       A/D.IN>ARRAY(DMA)          \ store in DATA.ARRAY


BEGIN                   \ !!!!!!!!!!!!!  for timing
?DMA.ACTIVE NOT                 \ !!!!!!!!!!!!
UNTIL                   \ !!!!!!!!!!!!


;

: CONVERT
   DATA.ARRAY                   \ recall A/D output
   -10 10 A/D.SCALE             \ convert to voltage
   VOLT.ARRAY :=                \    and store
;
\
\ harmonic analysis

\ LOAD.OVERLAY WAVEOPS.SOV          \ contains FFT routine
\       !!!!!!!!!!!! or configure in !!!!!!!!!!!!!!!


: HANNING.WINDOW  \ create Hanning window; allows timing discrepancy with
          \       reduced leakage to harmonics

       \ create index array, 0 - (N-1), multiple channels
   dp.real dim[ samples.per.channel ] unnamed.array
   SAMPLES.PER.CHANNEL RAMP 1. -
   over :=

       \ Hanning function on index array
```

```
   rad
   2 * pi * samples.per.channel 1 - / cos -1 * 1 +
   deg
   hann.window.array :=
;

: HARMONICS                        \ using FFT  routine
                                   \ multiple channel version
  NO.CHANNELS 1 +   1  DO          \
   SIGNAL.ARRAY                    \ 2-dim array (data x channel#)
   XSECT[ ! , I ]                  \ select I'th column (channel)
   hann.window.array *             \ Hanning window
   DP.REAL FFT FFT                 \ invoke FFT package

   \ plot spectrum                 !!!!!!!!!!!!!!!!!!!!!!!!!!!!
\    dup smag 1 + log sub[ 1 , samples.per.channel 2 / ]   \ !!!!!!!!!!!
                                   \ !!!!!!!!!!!
\    y.auto.plot                   \ !!!!!!!!!!!

   SUB[ NCYC 1 + , NO.HARMONICS , NCYC ]  \ select 60 Hs, harmonics

   2. *  SAMPLES.PER.CHANNEL /            \ normalisation(pts)
   1.414214 /                      \ peak -> rms
   DUP  FOURIER.COEFF     XSECT[ ! , I ] :=      \ complex coefficients
   DUP  ZMAG  FOURIER.MAG   XSECT[ ! , I ] :=      \ magnitudes
   DEG  ZARG  FOURIER.PHASE  XSECT[ ! , I ] :=     \ phases
  LOOP
;

: RELATIVE.PHASE       \ adjust phases for random sampling start time
                       \ adjust phase by subtracting
                       \ harmonic# * phase displacement
   FOURIER.PHASE XSECT[ 1 , 1 ]
   PHASE.REF :=                    \ select 60 Hs volts as phase ref
   DP.REAL DIM[ NO.CHANNELS , NO.HARMONICS ] UNNAMED.ARRAY
   NO.HARMONICS RAMP               \ create array with harmonic# for
   OVER :=                         \   all channels
   TRANS[ 1 , 2 ]
   PHASE.REF * FOURIER.PHASE SWAP -    \ multiply by phase displ. and
   FOURIER.PHASE :=                \     subtract from phase array
;

: PHASE.RANGE             \ put phases in -180 to +180 deg range
   FOURIER.PHASE
   180 +                  \ shift range up:   (-180,180) -> (0,360)
   360 MODULO             \ rectify out of range phases
   180 -                  \ shift range down: (0,360) -> (-180,180)
   FOURIER.PHASE :=
;

: DISPLAY                  \ simple printout
   CR ." AMPLITUDES ARE : " CR
   FOURIER.MAG .
   CR ." PHASES ARE : " CR
   FOURIER.PHASE .
;

: SYNTHESIZE            \ dual frequency cosine, all channel synthesis

   dp.real dim[ no.channels , samples.per.channel ] unnamed.array

   SAMPLES.PER.CHANNEL RAMP 1. -
   over :=  trans[ 1 , 2 ]

\    DUP                \ !!!!!!!!!!!!!!!
   2. * NCYC * PI *  ?conversion.delay * sampl.delay /
   SAMPLES.PER.CHANNEL /
   rad    COS 512 *  deg

\    SWAP               \ !!!!!!!!!!!!!!!
\    2. * 3. * NCYC * PI *  ?conversion.delay * sampl.delay /   \ add 3d harmon
\    SAMPLES.PER.CHANNEL /
\    rad    COS 512 *  deg
\    +
   2049.2 +  DATA.ARRAY :=         \ add offset and store
;

: PHASE.ZERO
   NO.CHANNELS 1 + 1 DO
     NOISE.BASE
     FOURIER.MAG   XSECT[ ! , I ]   [<]    \ is NOISE.BASE < MAGNITUDE ?
                                   \ (create 1/0 filter for Y/N)
     FOURIER.PHASE XSECT[ ! , I ]   *
```

```
    FOURIER.PHASE XSECT[ I , I ]  :=     \ if not, phase = 0
  LOOP
;

: PHASE.SHIFT

  \ adjust phases for sampling intervals - uniform interleaving
  \ phase adjustment depends on harmonic number, position in channel sequence
  \ correction is -( chan# - 1 )*( harm# )*( conv.delay )*21.6  deg


  DP.REAL DIM[ NO.HARMONICS , NO.CHANNELS ] UNNAMED.ARRAY \ fill channel array
  NO.CHANNELS RAMP  1 -
  OVER :=

  DP.REAL DIM[ NO.CHANNELS , NO.HARMONICS ] UNNAMED.ARRAY \ fill harmon. array
  NO.HARMONICS RAMP
  OVER :=
  TRANS[ 1 , 2 ]

  *                        \ multiply channel, harmonic arrays
  ?CONVERSION.DELAY * 21.6 *     \ numeric factor for 60 Hz, degrees, msec
  -1 * FOURIER.PHASE +           \ subtract from phase array
  FOURIER.PHASE :=               \ array of adjusted phases
;


: SENSOR.UNITS           \ sensor calibration: volts -> physical units
\   1. SENSOR.CALIBN :=    \ test patch DISABLED
  VOLT.ARRAY SENSOR.CALIBN *     \ voltage times calibration factor for
  SIGNAL.ARRAY :=                \          each channel
;


: PHYSICAL.UNITS          \ sensor calibration: volts -> physical units
  FOURIER.MAG
  SIGNAL.MAG :=          \ no-conversion patch
;


: FIELD.MAGNITUDE              \ Pythagorean combination only
  SIGNAL.MAG  SUB[ 5 , 3 ]   \ select orthogonal coil set
  DUP  TRANS[ 1 , 2 ]          \ align indices
  << * | + >>               \ inner product for each harmonic
  DIAG[ 1 , 1 ]             \ collapse on harmonic index
  SQRT
  SD.HARMONIC.MAGNITUDES :=   \ store results
;



: SAMPLE
\   LOAD.OVERLAY WAVEOPS.SOV         !!!!!!!!!!!!!!!!!!
  INIT
  DT2825.ACQUIRE


  CONVERT             \ digital to voltage conversion
  SENSOR.UNITS
  HANNING.WINDOW          \ create window function
  HARMONICS           \ fourier analysis of sampled signals
  RELATIVE.PHASE        \ refer phases to voltage fundamental
  PHASE.SHIFT         \ adjust phases for sample intervals
  PHASE.ZERO           \ ignore phases of harmonics < specified
                 \
  PHASE.RANGE          \ return phases to range (-180,180) deg
  PHYSICAL.UNITS       \ sensor calibration factors: volts -> units
  FIELD.MAGNITUDE
\   DISPLAY
;


: TEST.SYNTH
\   LOAD.OVERLAY WAVEOPS.SOV         !!!!!!!!!!!!!!!!!!
  INIT
\   DT2825.ACQUIRE       \ temporarily disabled !!!!!!!!!!!

  synthesise          \ !!!!!!!!!!!!!!!!!!!!

  CONVERT             \ digital to voltage conversion
  SENSOR.UNITS
  HANNING.WINDOW          \ create window function
  HARMONICS           \ fourier analysis of sampled signals
  RELATIVE.PHASE        \ refer phases to voltage fundamental
```

```
\   PHASE.SHIFT          \ adjust phases for sample intervals
\   PHASE.ZERO           \ ignore phases of harmonics < specified
amplitude
    PHASE.RANGE          \ return phases to range (-180,180) deg
    PHYSICAL.UNITS       \ sensor calibration factor
    DISPLAY
;
```

FILE SAMPLB

```
\ !!!!!!!!!!!!!!!!!! BEGIN MENU SECTION !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

\ for use with menus in this module: real-time display
XEQ MENU1                \ allows return to main menu
XEQ MENU2                \ allows return to REAL.TIME.DISPLAY menu
XEQ MENU3                \ allows return to SINGLE.CHANNEL selection menu
XEQ MENU4                \ allows return to SELECT.SINGLE.VIEW menu
XEQ MENU5                \ allows return to SINGLE.WAVE menu
XEQ MENU6                \ allows return to SINGLE.SPECTRUM menu

\ for integration with acquisition-storage and recall-display modules
XEQ RUN1                 \ allows transfer to File-based acquisition
XEQ RUN2                 \ allows transfer to File-recall display

INTEGER SCALAR CHAN.VIEWED

DIM[ 12 , 30 ] STRING.ARRAY CHAN.NAME   \ text for channel identification

   " VOLTAGE REFERENCE     "   CHAN.NAME "[ 1 ] ":=
   " CURRENT PROBE 3       "   CHAN.NAME "[ 2 ] ":=
   " CURRENT PROBE 2       "   CHAN.NAME "[ 3 ] ":=
   " CURRENT PROBE 1       "   CHAN.NAME "[ 4 ] ":=
   " MAGNETIC FIELD: X COIL "  CHAN.NAME "[ 5 ] ":=
   " MAGNETIC FIELD: Y COIL "  CHAN.NAME "[ 6 ] ":=
   " MAGNETIC FIELD: Z COIL "  CHAN.NAME "[ 7 ] ":=
   " AUXILIARY COIL        "   CHAN.NAME "[ 8 ] ":=
   " TOTAL MAGNETIC FIELD  "   CHAN.NAME "[ 9 ] ":=

DIM[ 12 , 15 ] STRING.ARRAY UNITS.NAME   \ text for graphics display

   " (VOLTS)     "   UNITS.NAME  "[ 1 ]  ":=
   " (AMPERES)   "   UNITS.NAME  "[ 2 ]  ":=
   " (AMPERES)   "   UNITS.NAME  "[ 3 ]  ":=
   " (AMPERES)   "   UNITS.NAME  "[ 4 ]  ":=
   " (MICROTESLA)"   UNITS.NAME  "[ 5 ]  ":=
   " (MICROTESLA)"   UNITS.NAME  "[ 6 ]  ":=
   " (MICROTESLA)"   UNITS.NAME  "[ 7 ]  ":=
   " (MICROTESLA)"   UNITS.NAME  "[ 8 ]  ":=
   " (MICROTESLA)"   UNITS.NAME  "[ 9 ]  ":=

: VU.SINGLE                  \ creates plotting window for wave, spectrum
   0 0.40 VUPORT.ORIG        \ origin of plotting window
   1 0.50 VUPORT.SIZE        \ size of plotting window
   VUPORT.CLEAR
;

: VU.LEFT
   0    0.4 VUPORT.ORIG
   0.45 0.5 VUPORT.SIZE
   VUPORT.CLEAR
;

: VU.RIGHT
   0.5  0.4 VUPORT.ORIG
   0.45 0.5 VUPORT.SIZE
   VUPORT.CLEAR
;

   0  0   0  35  WINDOW {VOLTS.WINDOW}        \ text window for signal ampl
   0  40  0  70  WINDOW {CHANNEL.WINDOW}      \ signal channel indicator
   17 0   17 79  WINDOW {MENU.NAME.WINDOW}     \ menu title
   18 0   24 79  WINDOW {INSTRUCTIONS.WINDOW}  \ menu window
   1  0   1  79  WINDOW {DISPLAY.TYPE.WINDOW}  \ display content identifier

: INIT.WINDOWS
   {VOLTS.WINDOW}        SCREEN.CLEAR
   {CHANNEL.WINDOW}      SCREEN.CLEAR
   {MENU.NAME.WINDOW}    SCREEN.CLEAR
   {INSTRUCTIONS.WINDOW} SCREEN.CLEAR
   {DISPLAY.TYPE.WINDOW} SCREEN.CLEAR
;

STORE.FUNCTION.KEYS      \ store settings before menus use f-keys

   XY:WORLD SPECTRUM.PLOT.WORLD
   XY:WORLD WAVE.PLOT.WORLD
```

```
: PLOT.WAVE
   SIGNAL.ARRAY XSECT[ ! , CHAN.VIEWED 1 + ]   \ specify a channel to plot
                          \ column = channel + 1
   dim[ samples.per.channel ] reshape     \ reconcile dimensions

   WAVE.PLOT.WORLD
   VU.SINGLE
   Y.AUTO.PLOT
;

: SINGLE.WAVE                          \ acquire and plot a waveform
   SAMPLE

   PLOT.WAVE


   {MENU.NAME.WINDOW}  SCREEN.CLEAR

   {INSTRUCTIONS.WINDOW} SCREEN.CLEAR
   ." <F1> TO REPEAT, <F2> TO LEAVE , <F3> FOR SPECTRUM "

   CLEAR.FUNCTION.KEYS
   F1 FUNCTION.KEY.DOES MENU5
   F2 FUNCTION.KEY.DOES MENU4
   F3 FUNCTION.KEY.DOES MENU6
   INTERPRET.KEYS                      \ await further instructions
;

: REPETITIVE.WAVE        \ continuously acquire and plot waveforms

   {MENU.NAME.WINDOW} SCREEN.CLEAR

   {INSTRUCTIONS.WINDOW} SCREEN.CLEAR
   ." PRESS ANY KEY TO HALT "

   CLEAR.FUNCTION.KEYS

   BEGIN
     SAMPLE
     SIGNAL.ARRAY XSECT[ ! , CHAN.VIEWED 1 + ]   \ specify a channel to plot
                          \ column = channel + 1
     dim[ samples.per.channel ] reshape
     VU.SINGLE
     Y.AUTO.PLOT
     3000 MSEC.DELAY    \ 3 seconds delay until next acquire
     ?KEY
   UNTIL

   MENU4                \ go to SELECT.SINGLE.VIEW
;


: PLOT.SPECTRUM
   SIGNAL.MAG XSECT[ ! , CHAN.VIEWED 1 + ]
   DIM[ NO.HARMONICS ] RESHAPE
   .0001 +        \ to avoid divide by zero error
   VU.LEFT
   SPECTRUM.PLOT.WORLD
   vertical logarithmic                \ logarithmic y-scale
   normal.coords .02 .05 label.scale.point    \ move y-scale factor
   scale.deltas[ 1. , 5. , 10. ]               \ restrict tick assignments
   dup [[max dup 1000 / swap vertical world.set  \ autorange, 3 decades
   0. no.harmonics horisontal world.set
   horizontal label.scale.off          \ suppress x-axis scale
   xy.axis.plot
   0.1 bar.width
   y.data.bar


   FOURIER.PHASE XSECT[ ! , CHAN.VIEWED 1 + ]
   DIM[ NO.HARMONICS ] RESHAPE
   VU.RIGHT
   vertical linear
   -200.  200.          VERTICAL   WORLD.SET
     0.  NO.HARMONICS   HORIZONTAL WORLD.SET
   XY.AXIS.PLOT
   Y.DATA.PLOT
;


: SINGLE.SPECTRUM

   SAMPLE

   VUPORT.CLEAR
```

```
   PLOT.SPECTRUM

   {MENU.NAME.WINDOW} SCREEN.CLEAR
   CR    ."          SPECTRUM "
   UNITS.NAME "[ CHAN.VIEWED 1 + ] "TYPE
   ."          PHASE (60 HZ) = "
   FOURIER.PHASE [ 1 , CHAN.VIEWED 1 + ]   . ." DEG "

   {INSTRUCTIONS.WINDOW} SCREEN.CLEAR
   CR CR CR CR ." <F1> TO REPEAT, <F2> TO LEAVE, <F3> FOR WAVEFORM "

   CLEAR.FUNCTION.KEYS
   F1 FUNCTION.KEY.DOES MENU6
   F2 FUNCTION.KEY.DOES MENU4
   F3 FUNCTION.KEY.DOES MENU5
   INTERPRET.KEYS
;

: REPETITIVE.SPECTRUM

   {MENU.NAME.WINDOW} SCREEN.CLEAR

   {INSTRUCTIONS.WINDOW} SCREEN.CLEAR
   CR CR ." PRESS ANY KEY TO HALT "

   CLEAR.FUNCTION.KEYS
   VUPORT.CLEAR

   BEGIN
      SAMPLE

      PLOT.SPECTRUM


      3000 MSEC.DELAY
      ?KEY
   UNTIL

   MENU4
;

: SELECT.SINGLE.VIEW

   CLEAR.FUNCTION.KEYS

   {CHANNEL.WINDOW}   SCREEN.CLEAR      \ indicate signal channel

     CHAN.NAME "[ CHAN.VIEWED 1 + ] "TYPE \ text in string array

   {MENU.NAME.WINDOW}  SCREEN.CLEAR     \ indicate menu type

     ." SELECT DISPLAY TYPE "


   {INSTRUCTIONS.WINDOW} SCREEN.CLEAR  \ menu of choices

     ." <F1> TIME-DOMAIN WAVEFORM - SINGLE SHOT "
   CR ." <F2>                   - REPETITIVE "
   CR ." <F3> SPECTRUM          - SINGLE SHOT "
   CR ." <F4>                   - REPETITIVE "
   CR ." <F5>                   - CRAWL "
   CR ." <F9>  SELECT ANOTHER CHANNEL            <F10> MAIN MENU "


   F1  FUNCTION.KEY.DOES SINGLE.WAVE
   F2  FUNCTION.KEY.DOES REPETITIVE.WAVE
   F3  FUNCTION.KEY.DOES SINGLE.SPECTRUM
   F4  FUNCTION.KEY.DOES REPETITIVE.SPECTRUM
   F5  FUNCTION.KEY.DOES BELL
   F9  FUNCTION.KEY.DOES MENU3
   F10 FUNCTION.KEY.DOES MENU2

   F6 ALT FUNCTION.KEY.DOES SCREEN.PRINT   \ SHADOW FUNCTION !!!!!!!!!!!!!!

   INTERPRET.KEYS
;

: CHANNEL.SELECT.0    0 CHAN.VIEWED :=     SELECT.SINGLE.VIEW ;
: CHANNEL.SELECT.1    1 CHAN.VIEWED :=     SELECT.SINGLE.VIEW ;
: CHANNEL.SELECT.2    2 CHAN.VIEWED :=     SELECT.SINGLE.VIEW ;
: CHANNEL.SELECT.3    3 CHAN.VIEWED :=     SELECT.SINGLE.VIEW ;
: CHANNEL.SELECT.4    4 CHAN.VIEWED :=     SELECT.SINGLE.VIEW ;
: CHANNEL.SELECT.5    5 CHAN.VIEWED :=     SELECT.SINGLE.VIEW ;
: CHANNEL.SELECT.6    6 CHAN.VIEWED :=     SELECT.SINGLE.VIEW ;
```

```
: CHANNEL.SELECT.7    7 CHAN.VIEWED :=      SELECT.SINGLE.VIEW  ;

: SELECT.TOTAL.FIELD

   8 CHAN.VIEWED :=

   {CHANNEL.WINDOW}  SCREEN.CLEAR      \ indicate signal channel

     CHAN.NAME "[ CHAN.VIEWED 1 + ] "TYPE \ text in string array


   {MENU.NAME.WINDOW} SCREEN.CLEAR

   {INSTRUCTIONS.WINDOW} SCREEN.CLEAR

   CR CR CR ." PRESS ANY KEY TO HALT "
   CLEAR.FUNCTION.KEYS

   BEGIN
      SAMPLE
      SD.HARMONIC.MAGNITUDES
      .0001 +       \ to avoid divide by zero error
      VU.RIGHT SCREEN.CLEAR
      VU.LEFT
      SPECTRUM.PLOT.WORLD
      vertical logarithmic                \ logarithmic y-scale
      normal.coords .02 .05 label.scale.point    \ move y-scale factor
      scale.deltas[ 1. , 5. , 10. ]              \ restrict tick assignments
      dup []max dup 1000 / swap vertical world.set  \ autorange, 3 decades
      0. no.harmonics horisontal world.set
      horizontal label.scale.off          \ suppress x-axis scale
      xy.axis.plot
      0.1 bar.width
      y.data.bar


      3000 MSEC.DELAY
      ?KEY
   UNTIL

   MENU3

;

: CHANNEL.MENU

   CR ." <F1> VOLTAGE REFERENCE     <F2>  CURRENT 3 "
   CR ." <F3> CURRENT 2           <F4>  CURRENT 1          "
   CR ." <F5> FIELD X           <F6>  FIELD Y              "
   CR ." <F7> FIELD Z           <F8>  AUXILIARY COIL       "
   CR ." <F9> TOTAL FIELD         <F10> RETURN                 "
;

: SINGLE.CHANNEL

   {CHANNEL.WINDOW} SCREEN.CLEAR


   {MENU.NAME.WINDOW} SCREEN.CLEAR

     ." SINGLE-CHANNEL MONITORING "

   {INSTRUCTIONS.WINDOW} SCREEN.CLEAR

   CHANNEL.MENU


   CLEAR.FUNCTION.KEYS

   F1  FUNCTION.KEY.DOES CHANNEL.SELECT.0    \ ADC board naming convention
   F2  FUNCTION.KEY.DOES CHANNEL.SELECT.1    \  channels 0 - 7
   F3  FUNCTION.KEY.DOES CHANNEL.SELECT.2
   F4  FUNCTION.KEY.DOES CHANNEL.SELECT.3
   F5  FUNCTION.KEY.DOES CHANNEL.SELECT.4
   F6  FUNCTION.KEY.DOES CHANNEL.SELECT.5
   F7  FUNCTION.KEY.DOES CHANNEL.SELECT.6
   F8  FUNCTION.KEY.DOES CHANNEL.SELECT.7
   F9  FUNCTION.KEY.DOES SELECT.TOTAL.FIELD
   F10 FUNCTION.KEY.DOES MENU2

   INTERPRET.KEYS



;
```

```
: MULTI.CHANNEL
   BELL
;

: DATA.LOGGING
   RUN1                \ transfers to file-based routines, below
;

: DATA.DISPLAY
   RUN2                \ transfers to file-recall and display, below
;

: STANDBY
   BELL
;

: MAIN.MENU

   {CHANNEL.WINDOW} SCREEN.CLEAR

   {MENU.NAME.WINDOW} SCREEN.CLEAR


   ." MAIN MENU "

   {INSTRUCTIONS.WINDOW} SCREEN.CLEAR

      ." <F1>  REAL-TIME DISPLAY - SINGLE CHANNEL"     \ instructions
   CR \ ." <F2>                - MULTIPLE CHANNELS"  !!!!!!!!!!
   CR ." <F3>  DATA-LOGGING MODE "
   CR ." <F4>  FILE-DISPLAY MODE "
   CR
   CR ." <F10> QUIT "

   CLEAR.FUNCTION.KEYS              \ each menu has unique key uses

   F1  FUNCTION.KEY.DOES  SINGLE.CHANNEL   \ function keys invoke menu
   F2  FUNCTION.KEY.DOES  MULTI.CHANNEL
   F3  FUNCTION.KEY.DOES  DATA.LOGGING      \     choices
   F4  FUNCTION.KEY.DOES  DATA.DISPLAY
   F5  FUNCTION.KEY.DOES  STANDBY
   F10 FUNCTION.KEY.DOES  BYE

   INTERPRET.KEYS                   \ wait for selection
;

SCREEN.CLEAR

: INSTALL.MENUS

   INSTALL   MAIN.MENU          IN  MENU1
   INSTALL   MAIN.MENU          IN  MENU2
   INSTALL   SINGLE.CHANNEL     IN  MENU3
   INSTALL   SELECT.SINGLE.VIEW IN  MENU4
   INSTALL   SINGLE.WAVE        IN  MENU5
   INSTALL   SINGLE.SPECTRUM    IN  MENU6

;

: GO
   SCREEN.CLEAR
   GRAPHICS.DISPLAY
   VU.SINGLE
   SCREEN.CLEAR
   INSTALL.MENUS        \ makes returns possible
   INIT.WINDOWS
   MAIN.MENU
;
```

```
FILE SAMPLC

\ FILE SET-UP SECTION FOR DATA.LOGGING

\ XEQ NO.EVENTS

INTEGER SCALAR NO.ARRAYS
REAL SCALAR NO.EVENTS          \ patch for testing !!!!!!!!!
REAL DIM[ 1 ] ARRAY INTERVAL
REAL DIM[ 1 ] ARRAY DURATION

   8192 NO.HARMONICS / NO.CHANNELS /
   NO.EVENTS :=          \ maximum number of events to fill array

REAL DIM[ NO.EVENTS , 2 , NO.HARMONICS , NO.CHANNELS ]
   ARRAY CAPTURED.EVENTS.SEQUENCE
REAL DIM[ NO.EVENTS , NO.CHANNELS 1 + ] ARRAY DISPLAY.BUFFER
25 STRING NAME.FOR.FILE
64 STRING IDENTIFIER
64 STRING LOCATION
3  STRING SEQUENCE.NO
64 STRING REMARKS.1
64 STRING REMARKS.2

: IN.FILE.NAME
   CR CR ." 1. INPUT NAME FOR FILE :  "
   "INPUT NAME.FOR.FILE ":=
;

: IN.IDENT.NAME
   CR CR ." 2. INPUT IDENTIFYING NAME : "
   "INPUT IDENTIFIER ":=
;

: IN.LOCATION
   CR CR ." 3. INPUT LOCATION : "
   "INPUT LOCATION ":=
;

: IN.SEQUENCE.NO
   CR CR ." 4. INPUT SEQUENCE # (XXX) : "
   "INPUT SEQUENCE.NO ":=
;

: IN.REMARKS
   CR CR ." 5. ADDITIONAL REMARKS (2 LINES) : "
   CR "INPUT REMARKS.1 ":=
   CR "INPUT REMARKS.2 ":=
;

: PRINT.FILE.DESCRIPTORS
   CR CR ." 1: NAME FOR FILE     :    " NAME.FOR.FILE "TYPE
   CR CR ." 2: IDENTIFYING NAME  :    " IDENTIFIER   "TYPE
   CR CR ." 3: LOCATION          :    " LOCATION     "TYPE
   CR CR ." 4: SEQUENCE #        :    " SEQUENCE.NO  "TYPE
   CR CR ." 5: ADDITIONAL REMARKS :   "
         CR  REMARKS.1 "TYPE
         CR  REMARKS.2 "TYPE
;

: CHANGE.FILE.DESCRIPTORS
   {DEF}  SCREEN.CLEAR
   PRINT.FILE.DESCRIPTORS
   CR CR CR ." CHANGE ANY INPUT DATA ? INPUT ITEM # OR RETURN "
   #INPUT

   IF                          \ if a numeric response
      CASE                     \ select item to change
      1 OF  IN.FILE.NAME   ENDOF
      2 OF  IN.IDENT.NAME  ENDOF
      3 OF  IN.LOCATION    ENDOF
      4 OF  IN.SEQUENCE.NO ENDOF
      5 OF  IN.REMARKS     ENDOF
      ENDCASE
   MYSELF                      \ repeat for further changes
   THEN


;


: INPUT.FILE.DESCRIPTORS
```

```
   IN.FILE.NAME
   IN.IDENT.NAME
   IN.LOCATION
   IN.SEQUENCE.NO
   IN.REMARKS
;

10 STRING INPUT.ANSWER

: INPUT.LOGGING.PARAMETERS

   {DEF} SCREEN.CLEAR
   CR ." INPUT SAMPLE INTERVAL (MINUTES) "
   CR ."       DEFAULT 0.25 MINUTES    "
   0.25 INTERVAL [ 1 ] :=
   #INPUT                    \ -> t/f if numeric input
   IF                      \ checks for valid input type
      INTERVAL [ 1 ] :=
   THEN

   CR ." INPUT RUN DURATION (HOURS) "
   CR ."       DEFAULT 24 HOURS    "
   24. DURATION [ 1 ] :=
   #INPUT
   IF
      DURATION [ 1 ] :=
   THEN

   INTERVAL [ 1 ] CR CR ." SAMPLE INTERVAL = "  .  ." MINUTES "
   DURATION [ 1 ] CR ." RUN DURATION    = "  .  ." HOURS "

   CR CR ." ACCEPT THESE VALUES ? Y/N "
   "INPUT  "DUP
   INPUT.ANSWER ":=          \ avoids mixed use of symbol stack (OVER,SWAP)
   " Y"  "=                  \ leaves T/F on symbol stack

   INPUT.ANSWER
   " y"  "=                \ leaves T/F next on symbol stack

   OR NOT                    \ Y or y -> False, anything else -> True

   IF
      SCREEN.CLEAR
      MYSELF                 \ repeat input step if not Y or y
   THEN
;

: FILE.SETUP

   FILE.TEMPLATE
      6 COMMENTS             \ file identifying descriptors
      REAL DIM[ 1 ] SUBFILE  \ to store INTERVAL, DURATION parameters
      2 TIMES
      \ date & time stamps, data array will be appended during run
   END

   NAME.FOR.FILE
   DEFER> FILE.CREATE

   NAME.FOR.FILE
   DEFER> FILE.OPEN

   1 IDENTIFIER    >COMMENT
   2 LOCATION      >COMMENT
   3 "DATE       >COMMENT
   4 SEQUENCE.NO   >COMMENT
   5 REMARKS.1     >COMMENT
   6 REMARKS.2     >COMMENT


   1 SUBFILE  INTERVAL  ARRAY>FILE
   2 SUBFILE  DURATION  ARRAY>FILE

;

INTEGER DIM[ 10 ] ARRAY DATE.STAMP
INTEGER DIM[ 10 ] ARRAY TIME.STAMP

: DATE.TIME.TO.FILE

   "DATE  DATE.STAMP  ">ARRAY  \ stored in subfiles as integer arrays
   "TIME  TIME.STAMP  ">ARRAY

   DATE.STAMP   APPEND.ARRAY>FILE
```

```
   TIME.STAMP   APPEND.ARRAY>FILE
;

INTEGER SCALAR DISPLAY.TOGGLE

: LOGGING.DISPLAY.ON
   1 DISPLAY.TOGGLE :=
   pckey drop ?drop
;

: LOGGING.DISPLAY.OFF
   0 DISPLAY.TOGGLE :=
   pckey drop ?drop
;

: END.LOGGING \ not programmed yet
   {MENU.NAME.WINDOW} SCREEN.CLEAR
;

INTEGER SCALAR CHANNEL.A
INTEGER SCALAR CHANNEL.B
INTEGER SCALAR N.HARMON      \ select harmonic for display
INTEGER SCALAR DATA.TYPE     \ select magnitude or phase display

: CHANNEL.DEFAULTS
   4 CHANNEL.A :=                \ current probe 1
   9 CHANNEL.B :=                \ total magnetic field
   1 N.HARMON  :=
   1 DATA.TYPE :=
;

: DISPLAY.CRAWL

   {DISPLAY.TYPE.WINDOW} SCREEN.CLEAR
   ." HARMONIC NUMBER = "  N.HARMON .

   2 DATA.TYPE =
   IF  ."   PHASE DISPLAY "
   ELSE ."              " THEN

   NO.EVENTS   INTERVAL [ 1 ]  *
   ."  FRAME LENGTH = " . ." MIN"

   {MENU.NAME.WINDOW}               \ displays channel names
   SCREEN.CLEAR

   CHAN.NAME  "[ CHANNEL.A ] "TYPE
   UNITS.NAME "[ CHANNEL.A ]        \ field or current units
   2 DATA.TYPE =
   IF                          \ unless phase display
      "DROP
      " (DEGREES)  "                \ then degrees
   THEN
   "TYPE


   ."    "

   CHAN.NAME  "[ CHANNEL.B ] "TYPE
   UNITS.NAME "[ CHANNEL.B ]
   2 DATA.TYPE =
   IF
      "DROP
      " (DEGREES)  "
   THEN
   "TYPE


   DISPLAY.BUFFER XSECT[ ! , CHANNEL.A ]    \ left-hand display
   DIM[ NO.EVENTS ] RESHAPE
   VU.LEFT
   Y.AUTO.PLOT

   DISPLAY.BUFFER XSECT[ ! , CHANNEL.B ]    \ right-hand display
   DIM[ NO.EVENTS ] RESHAPE
   VU.RIGHT
   Y.AUTO.PLOT

;

: DISPLAY.not.CRAWL
   {MENU.NAME.WINDOW}
   SCREEN.CLEAR cr ." display crawl toggled off "
```

```
   VU.LEFT  SCREEN.CLEAR
   VU.RIGHT SCREEN.CLEAR

;


: END.LOGGING.SAVE
   CAPTURED.EVENTS.SEQUENCE
   APPEND.ARRAY>FILE
   FILE.CLOSE
   MENU1
;

: END.LOGGING.NO.SAVE
   FILE.CLOSE
   MENU1
;

INTEGER SCALAR CHANNEL.CODE

: WOGGLE
   PCKEY
   IF
      CASE
      59  OF 1 CHANNEL.CODE  !=   ENDOF  \ F1
      60  OF 2 CHANNEL.CODE  !=   ENDOF
      61  OF 3 CHANNEL.CODE  !=   ENDOF
      62  OF 4 CHANNEL.CODE  !=   ENDOF
      63  OF 5 CHANNEL.CODE  !=   ENDOF
      64  OF 6 CHANNEL.CODE  !=   ENDOF
      65  OF 7 CHANNEL.CODE  !=   ENDOF
      66  OF 8 CHANNEL.CODE  !=   ENDOF
      67  OF 9 CHANNEL.CODE  !=   ENDOF  \ F9
      68  OF END.LOGGING.SAVE     ENDOF  \ F10
      113 OF END.LOGGING.NO.SAVE  ENDOF  \ ALT F10
      ENDCASE
      ELSE DROP
   THEN
;

: CHECK.FOR.WOGGLE
   ?KEY          \ checks for input but does not empty keyboard buffer
   IF WOGGLE THEN
;

: TOGGLE
   DISPLAY.TOGGLE 1 +
   2 MODULO              \ 0  nterchanges with 1
   DISPLAY.TOGGLE !=
   PCKEY DROP ?DROP      \ removes input from keyboard buffer
;

: CHECK.FOR.TOGGLE
   ?KEY          \ checks for input but does not empty keyboard buffer
   IF TOGGLE THEN
;

: SYNCHRONIZE.IT
   SYNCHRONIZE

   ONERR: {DEF} SCREEN.CLEAR
      ." STOPPED AT ERROR - ked. " BREAK

;

DP.REAL SCALAR TAU.1
DP.REAL SCALAR TAU.2
REAL SCALAR SYNCH


: COACH.RIDE
   BEGIN
      REL.TIME  1000. /
      INTERVAL [ 1 ] 60 *  <        \ wait until past midnight
   UNTIL
   TAU.2  1000 * MSEC.DELAY         \ make up rest of interval
   SYNCH  SYNC.PERIOD               \ reset SYNCH timer
;


: ?CINDERELLA    \ checks if next sample is after midnight
```

```
  86400.  REL.TIME  1000. / -       \ units: seconds
  TAU.1 :=                         \ time to midnight
  INTERVAL [ 1 ] 60 *    TAU.1  -
  TAU.2 :=                         \ midnight to next sample
  TAU.1  INTERVAL [ 1 ] 60 *   <    \ last sample before midnight
                               \    has occurred?
  IF
    COACH.RIDE          \ wait past midnight & reset synch
  THEN
;



: DATA.LOGGING.CONTROL

  {DEF} SCREEN.CLEAR

  0 DISPLAY.TOGGLE :=          \ set for no display

  INTERVAL [ 1 ]  1000 * 60 *

  DUP SYNCH :=

  SYNC.PERIOD \ interval between captures, msec

  DURATION [ 1 ] 60 * NO.EVENTS / INTERVAL [ 1 ]  /

  NO.ARRAYS :=     \ number of arrays dumped to file during run duration

  CLEAR.FUNCTION.KEYS
  {INSTRUCTIONS.WINDOW} SCREEN.CLEAR

CR  ." input new display channels: left, followed promptly by right" CR
."    <F1>      <F2>      <F3>      <F4>      <F5>      <F6> " CR
." VOLTAGE REF   CURRENT 3   CURRENT 2   CURRENT 1   FIELD X    FIELD Y " CR
."    <F7>      <F8>      <F9> " CR
."  FIELD Z     AUXIL COIL   TOTAL FIELD

  CR ." <F10> ENDS RUN WITH DATA SAVE.   "
  ." ALT <F10> ENDS RUN (NO SAVE)"


NO.ARRAYS 1 + 1 DO

  0. CAPTURED.EVENTS.SEQUENCE :=     \ initialise storage array
  0. DISPLAY.BUFFER :=

  DATE.TIME.TO.FILE


  NO.EVENTS 1 +  1  DO
    SYNCHRONIZE.IT          \ INTERVAL sets repetition rate

    SAMPLE

    \ store samples sequentially in array
    FOURIER.MAG
     CAPTURED.EVENTS.SEQUENCE XSECT[ I , 1 , ! , ! ] :=
    FOURIER.PHASE
     CAPTURED.EVENTS.SEQUENCE XSECT[ I , 2 , ! , ! ] :=

    \ copy fundamental's magnitudes into array for display
    FOURIER.MAG SUB[ 1 , 1 ; 1 , NO.CHANNELS ]
     DISPLAY.BUFFER SUB[ I , 1 ; 1 , NO.CHANNELS ] :=

    \ include magnetic field magnitude (3-dim)
    3D.HARMONIC.MAGNITUDES [ 1 ]
     DISPLAY.BUFFER [ I , NO.CHANNELS 1 + ] :=

    {INSTRUCTIONS.WINDOW}
\    CR ." EVENT # " I .

\    CHECK.FOR.TOGGLE

\    DISPLAY.TOGGLE
\      CASE
\        1 OF  DISPLAY.CRAWL     ENDOF
\        0 OF  DISPLAY.NOT.CRAWL  ENDOF
\
\      ENDCASE

    CHANNEL.A     CHANNEL.CODE  :=    \ store CH.A
```

```
    CHECK.FOR.WOGGLE                \ change if an input
    CHANNEL.CODE   CHANNEL.A    :=    \ reset CH.A

    CHANNEL.B      CHANNEL.CODE :=    \ store CH.B
    CHECK.FOR.WOGGLE                \ change if an input
    CHANNEL.CODE   CHANNEL.B    :=    \ reset CH.B


    DISPLAY.CRAWL

    ?CINDERELLA         \ check if close to midnight

  LOOP
  CAPTURED.EVENTS.SEQUENCE
  APPEND.ARRAY>FILE

 LOOP
 FILE.CLOSE
;


: DATA.FILE.SETUP

   {DEF} SCREEN.CLEAR
   INPUT.FILE.DESCRIPTORS
   CHANGE.FILE.DESCRIPTORS
   INPUT.LOGGING.PARAMETERS
   FILE.SETUP

;

: RUN.DATA

   DATA.FILE.SETUP
   CHANNEL.DEFAULTS

   {DEF} SCREEN.CLEAR
   ." <F1> TO START DATA ACQUISITION "
   CR CR ." <F10> TO ABORT  "

   F1  FUNCTION.KEY.DOES DATA.LOGGING.CONTROL
   F10 FUNCTION.KEY.DOES MENU1

   INTERPRET.KEYS

   {DEF} SCREEN.CLEAR
   ." RUN ENDED SUCCESSFULLY"
   CR ." DATA STORED IN FILE :  "  NAME.FOR.FILE  "TYPE

;

INSTALL RUN.DATA IN RUN1   \ integrates file-based acquisition and
                           \  storage with real-time monitoring
```

FILE SAMPLD

```
\
\ PROGRAM TO READ BACK ASYST FILE CONTAINING CURRENT AND
\    MAGNETIC FIELD ARRAYS - 8 CHANNELS

\ MARCH 1, 1989

XEQ ERR.ROUTE.1

INTEGER SCALAR FRAME.NO
INTEGER SCALAR NO.ARRAYS.STORED
15 STRING FILE.DATE
15 STRING DATEBACK
15 STRING TIMEBACK


: READ.DATE.TIME.STAMPS
   FRAME.NO  3 *      SUBFILE
   DATE.STAMP FILE>ARRAY
   FRAME.NO  3 *  1 +  SUBFILE
   TIME.STAMP FILE>ARRAY
   DATEBACK DATE.STAMP ARRAY>"
   TIMEBACK TIME.STAMP ARRAY>"

;



: GET.FILE
   DEFER> FILE.OPEN
   ONERR:
      BELL
      CR CR NAME.FOR.FILE "TYPE  ."  DOES NOT EXIST "
      3000 MSEC.DELAY BELL
      ERR.ROUTE.1
;


: READ.FILE.SPEC

   1 FRAME.NO :=

   {DEF} SCREEN.CLEAR

   F10 FUNCTION.KEY.DOES MENU1
   ." NAME OF FILE TO READ ?" CR CR ." <F10> TO EXIT " CR

   "INPUT  "DUP
   NAME.FOR.FILE ":=

   GET.FILE

   \ READ COMMENTS

      1 COMMENT>  IDENTIFIER   ":=
      2 COMMENT>  LOCATION     ":=
      3 COMMENT>  FILE.DATE    ":=
      4 COMMENT>  SEQUENCE.NO  ":=
      5 COMMENT>  REMARKS.1    ":=
      6 COMMENT>  REMARKS.2    ":=

   \ READ DATA LOGGING PARAMETERS

      1 SUBFILE  INTERVAL   FILE>ARRAY
      2 SUBFILE  DURATION   FILE>ARRAY

   \ ASCERTAIN # OF DATA ARRAYS STORED

      FILE.CONTENTS   \ # comments, # subfiles
      SWAP  DROP      \ remove # comments from stack
      2 -             \ allow for interval, duration variables
      DUP             \ }
      3 MODULO -      \ }integer roundoff gives # of event arrays
      3 /             \ }
      NO.ARRAYS.STORED :=

   {DEF} SCREEN.CLEAR
   PRINT.FILE.DESCRIPTORS

   0 NO.ARRAYS.STORED =
   IF
      CR CR CR ." NO DATA STORED IN THIS FILE " CR CR
      FILE.CLOSE
```

```
   THEN
   CR CR CR CR ." PRESS ANY KEY TO CONTINUE "
   BEGIN
     ?KEY
   UNTIL
   PCKEY DROP ?DROP
   0 NO.ARRAYS.STORED =
   IF
     MYSELF
   THEN
;

: FILE.CONTENTS.PLOT

   READ.DATE.TIME.STAMPS
   FRAME.NO  3 *  2 +  SUBFILE
   CAPTURED.EVENTS.SEQUENCE    FILE>ARRAY   \ reads one array from file
   CAPTURED.EVENTS.SEQUENCE  XSECT[ ! , DATA.TYPE , N.HARMON , ! ]
     DISPLAY.BUFFER  SUB[ 1 , NO.EVENTS ; 1 , NO.CHANNELS ] :=
   CAPTURED.EVENTS.SEQUENCE  XSECT[ ! , 1 , N.HARMON , 5 ] DUP *
   CAPTURED.EVENTS.SEQUENCE  XSECT[ ! , 1 , N.HARMON , 6 ] DUP *  +
   CAPTURED.EVENTS.SEQUENCE  XSECT[ ! , 1 , N.HARMON , 7 ] DUP *  +
   SQRT
   DISPLAY.BUFFER XSECT[ ! , NO.CHANNELS 1 + ]  :=
   {DEF} SCREEN.CLEAR
   {VOLTS.WINDOW} SCREEN.CLEAR
   ." FILE "
   NAME.FOR.FILE "TYPE
   ."     "  DATEBACK "TYPE

   {CHANNEL.WINDOW} SCREEN.CLEAR
   TIMEBACK "TYPE
   ."   FRAME " FRAME.NO . ." /" NO.ARRAYS.STORED .

   {INSTRUCTIONS.WINDOW} SCREEN.CLEAR
   DISPLAY.CRAWL
;

: SCAN.LEFT
   FRAME.NO  1  >    \ check possibility of left move
   IF
     FRAME.NO  1 -  FRAME.NO  :=    \ decrement FRAME.NO
   THEN
   FILE.CONTENTS.PLOT
;
: SCAN.RIGHT
   FRAME.NO  NO.ARRAYS.STORED  <    \ check possibility of right move
   IF
     FRAME.NO  1 +  FRAME.NO  :=    \ increment FRAME.NO
   THEN
   FILE.CONTENTS.PLOT
;
: HOME.LEFT
   1 FRAME.NO :=
   FILE.CONTENTS.PLOT
;
: END.RIGHT
   NO.ARRAYS.STORED FRAME.NO :=
```

```
   FILE.CONTENTS.PLOT
;

: CHANNEL.HOP   \ translate key response to channel #

   CASE
      59 OF   1   CHANNEL.CODE :=   ENDOF    \ F1
      60 OF   2   CHANNEL.CODE :=   ENDOF
      61 OF   3   CHANNEL.CODE :=   ENDOF
      62 OF   4   CHANNEL.CODE :=   ENDOF
      63 OF   5   CHANNEL.CODE :=   ENDOF
      64 OF   6   CHANNEL.CODE :=   ENDOF
      65 OF   7   CHANNEL.CODE :=   ENDOF
      66 OF   8   CHANNEL.CODE :=   ENDOF
      67 OF   9   CHANNEL.CODE :=   ENDOF    \ F9
   ENDCASE
;

: SWITCH.CHANNELS

   {INSTRUCTIONS.WINDOW} SCREEN.CLEAR

   CR ." LEFT-HAND CHANNEL ?"
   CHANNEL.MENU

   BEGIN ?KEY UNTIL

   PCKEY
   CHANNEL.A CHANNEL.CODE :=
   IF CHANNEL.HOP ELSE DROP THEN
   CHANNEL.CODE CHANNEL.A :=

   SCREEN.CLEAR
   CR ." RIGHT-HAND CHANNEL ?"
   CHANNEL.MENU

   BEGIN ?KEY UNTIL

   PCKEY
   CHANNEL.B CHANNEL.CODE :=
   IF CHANNEL.HOP ELSE DROP THEN
   CHANNEL.CODE CHANNEL.B :=

   FILE.CONTENTS.PLOT

;

: DISP.MAGNITUDES
   1 DATA.TYPE :=
   1 N.HARMON  :=

   FILE.CONTENTS.PLOT
;

: DISP.HARMONICS
   {INSTRUCTIONS.WINDOW} SCREEN.CLEAR
   ." ENTER HARMONIC NUMBER "  #INPUT

   IF
      N.HARMON :=
\     ELSE DROP        \ how to handle non-valid input ?????
   THEN

   1 DATA.TYPE :=

   FILE.CONTENTS.PLOT
;

: DISP.PHASES
   2 DATA.TYPE :=
\   1 N.HARMON  :=    \ may be disabled to allow non-default option

   FILE.CONTENTS.PLOT
;

: ANOTHER.FILE
   FILE.CLOSE
   RUN2
;

: RETURN1
   FILE.CLOSE
```

```
   MENU1
;

: DIRECTIONS
   CASE
     75  OF   SCAN.LEFT      ENDOF    \ <-
     77  OF   SCAN.RIGHT     ENDOF    \ ->
     71  OF   HOME.LEFT      ENDOF    \ Home
     79  OF   END.RIGHT      ENDOF    \ End
     59  OF   SWITCH.CHANNELS  ENDOF     \ F1
     60  OF   DISP.MAGNITUDES  ENDOF     \ F2
     61  OF   DISP.HARMONICS   ENDOF     \ F3
     62  OF   DISP.PHASES     ENDOF    \ F4
     67  OF   ANOTHER.FILE    ENDOF    \ F9
     68  OF   RETURN1        ENDOF    \ F10

    109  OF   SCREEN.PRINT    ENDOF    \ ALT F6 - SHADOW FUNCTION
                              \ !!!!!!!!!!!!!!!!!!

   ENDCASE
;

: PLOT.FILE.CONTROL


   {INSTRUCTIONS.WINDOW} SCREEN.CLEAR

   CR CR ." CURSOR  <- , -> , HOME, END TO SCAN RECORD "
   CR CR ." <F1>  TO SWITCH CHANNELS "
   CR ." <F2>  FOR MAGNITUDES        <F9>  TO READ ANOTHER FILE "
   CR ." <F3>  FOR HARMONICS         <F10> TO EXIT              "
   CR ." <F4>  FOR PHASE             "

   BEGIN
     ?KEY
   UNTIL

   PCKEY
   IF
     DIRECTIONS
     ELSE DROP
   THEN

   MYSELF
;


: LOOK
   GRAPHICS.DISPLAY
   READ.FILE.SPEC
   CHANNEL.DEFAULTS
   FILE.CONTENTS.PLOT
   PLOT.FILE.CONTROL
;

INSTALL LOOK IN RUN2
INSTALL LOOK IN ERR.ROUTE.1
```

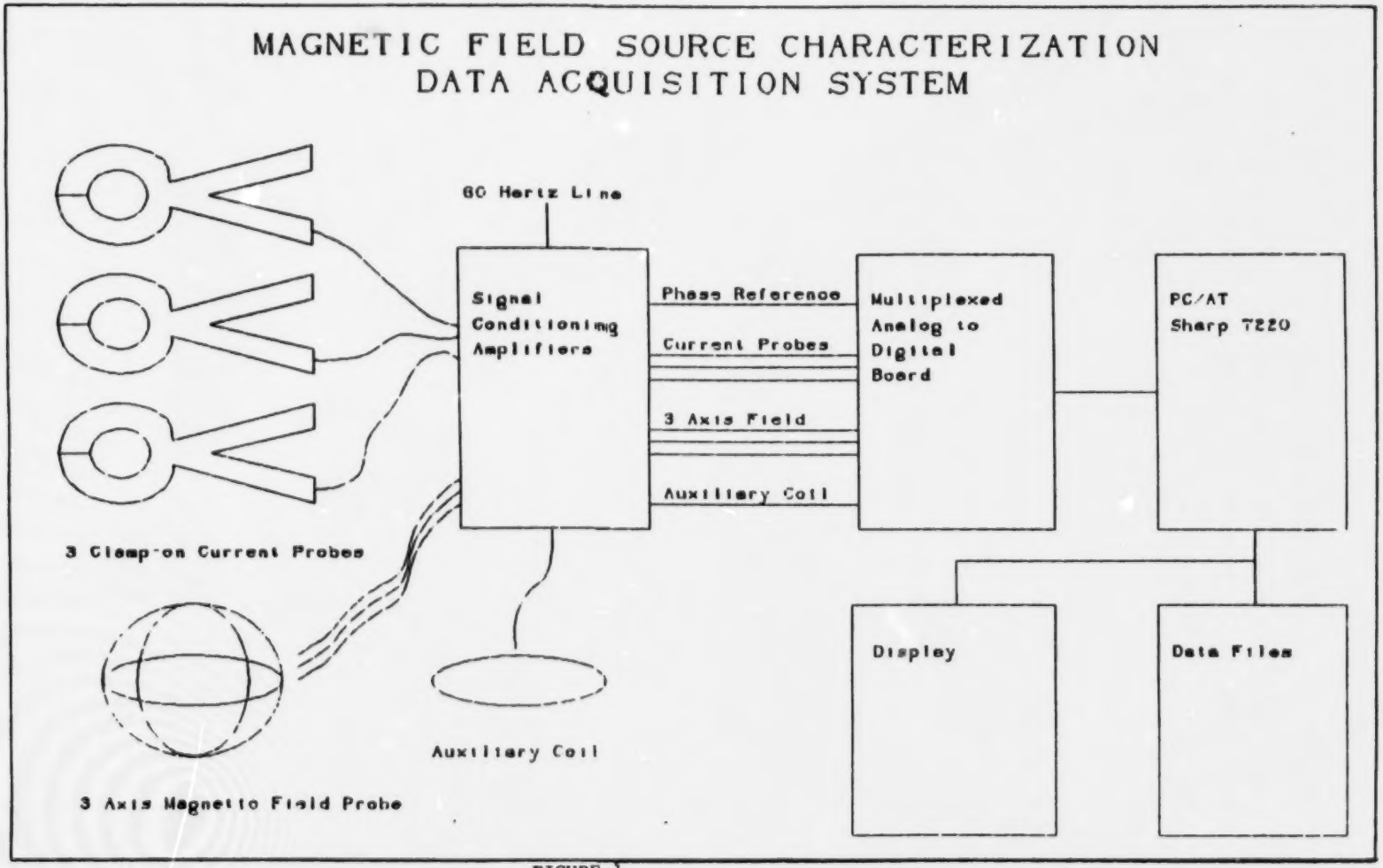


FIGURE 1

SYSTEM COMPONENTS

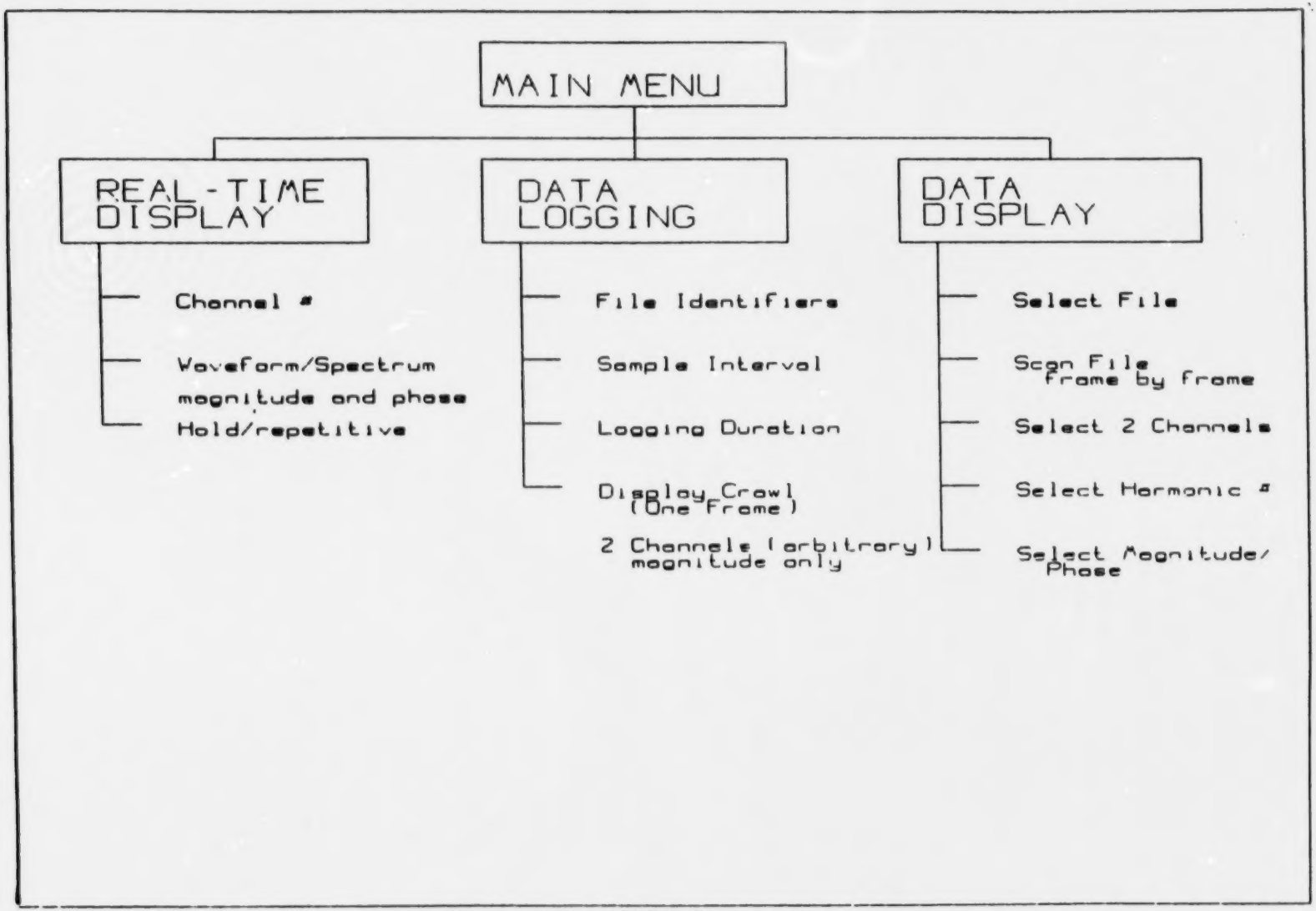


FIGURE 2

SOFTWARE FUNCTIONS

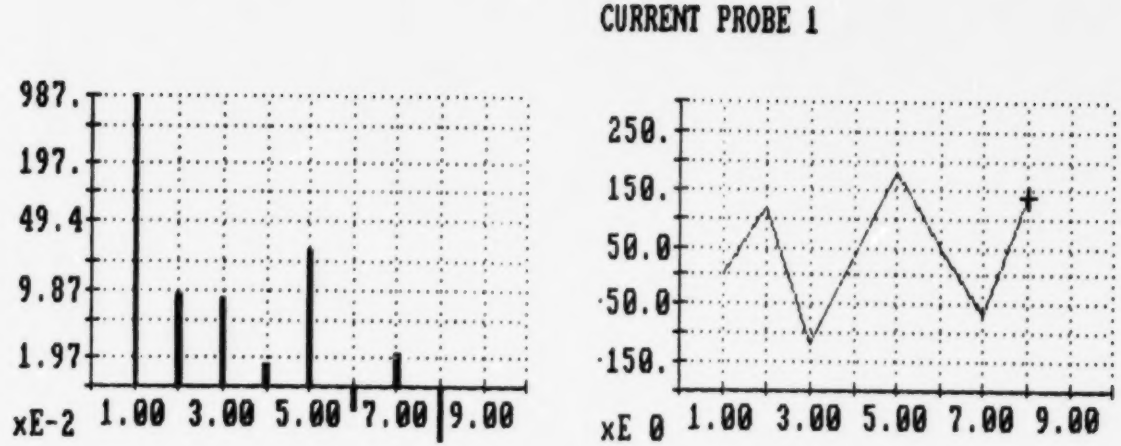


FIGURE 3

REAL-TIME DISPLAY

FIGURE 4

DISPLAY OF LOGGED DATA FILE, SHOWING MAGNITUDES

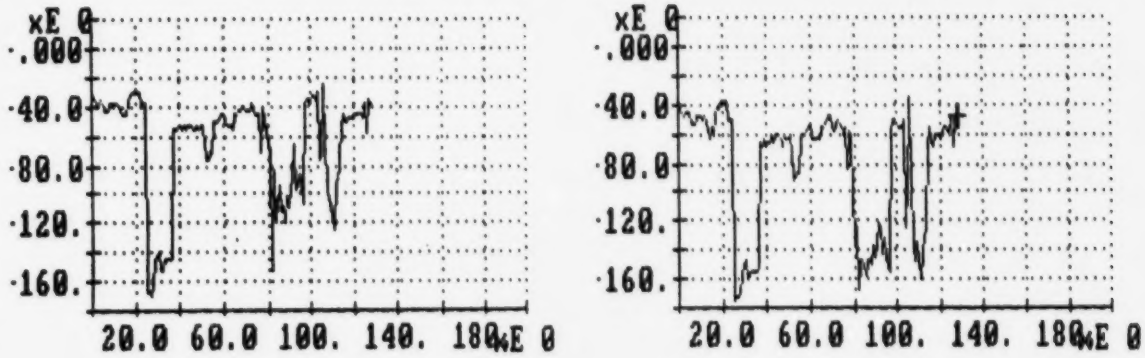


FIGURE 5

DISPLAY OF LOGGED DATA FILE, SHOWING PHASES

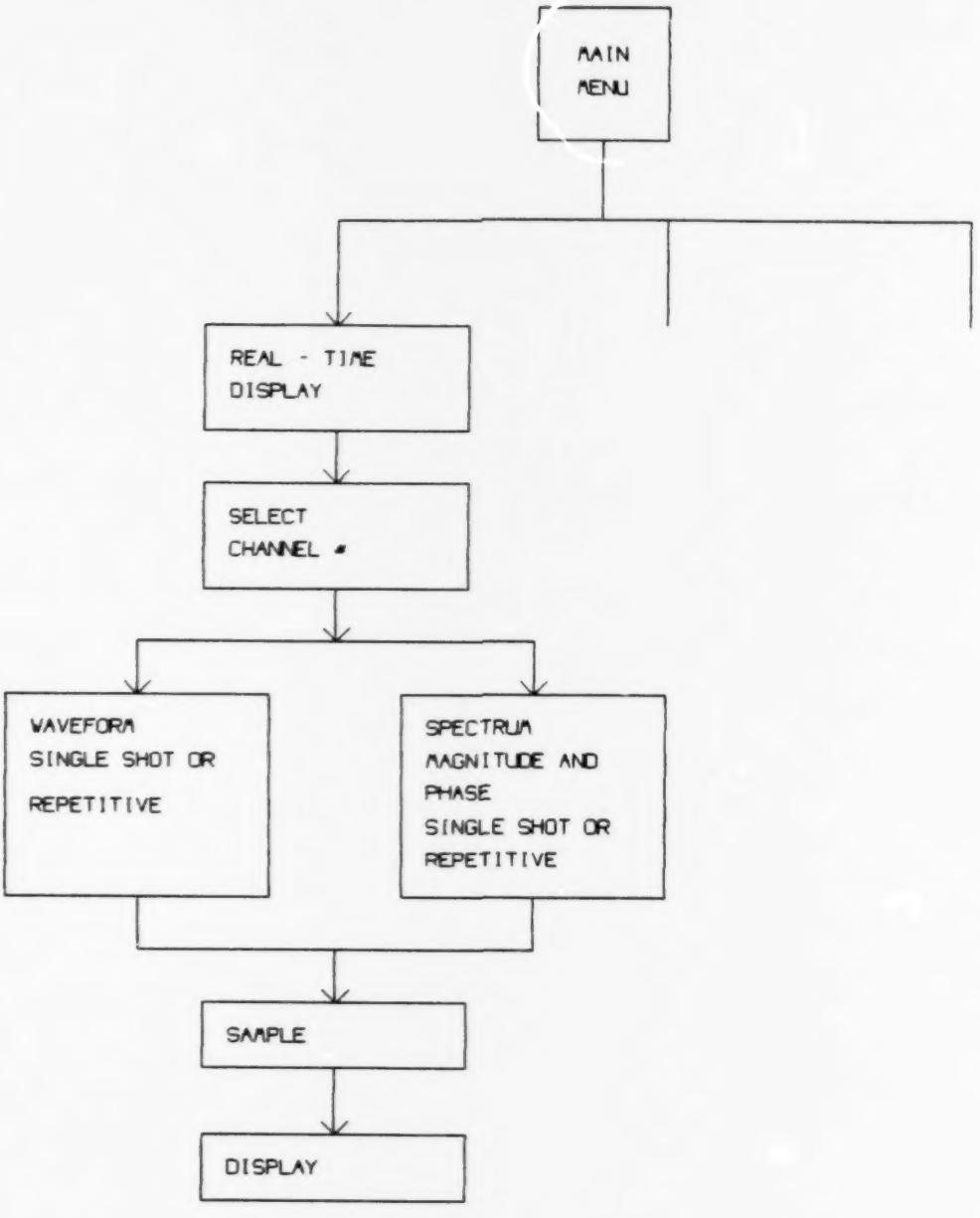


FIGURE A1

REAL-TIME DISPLAY MENU

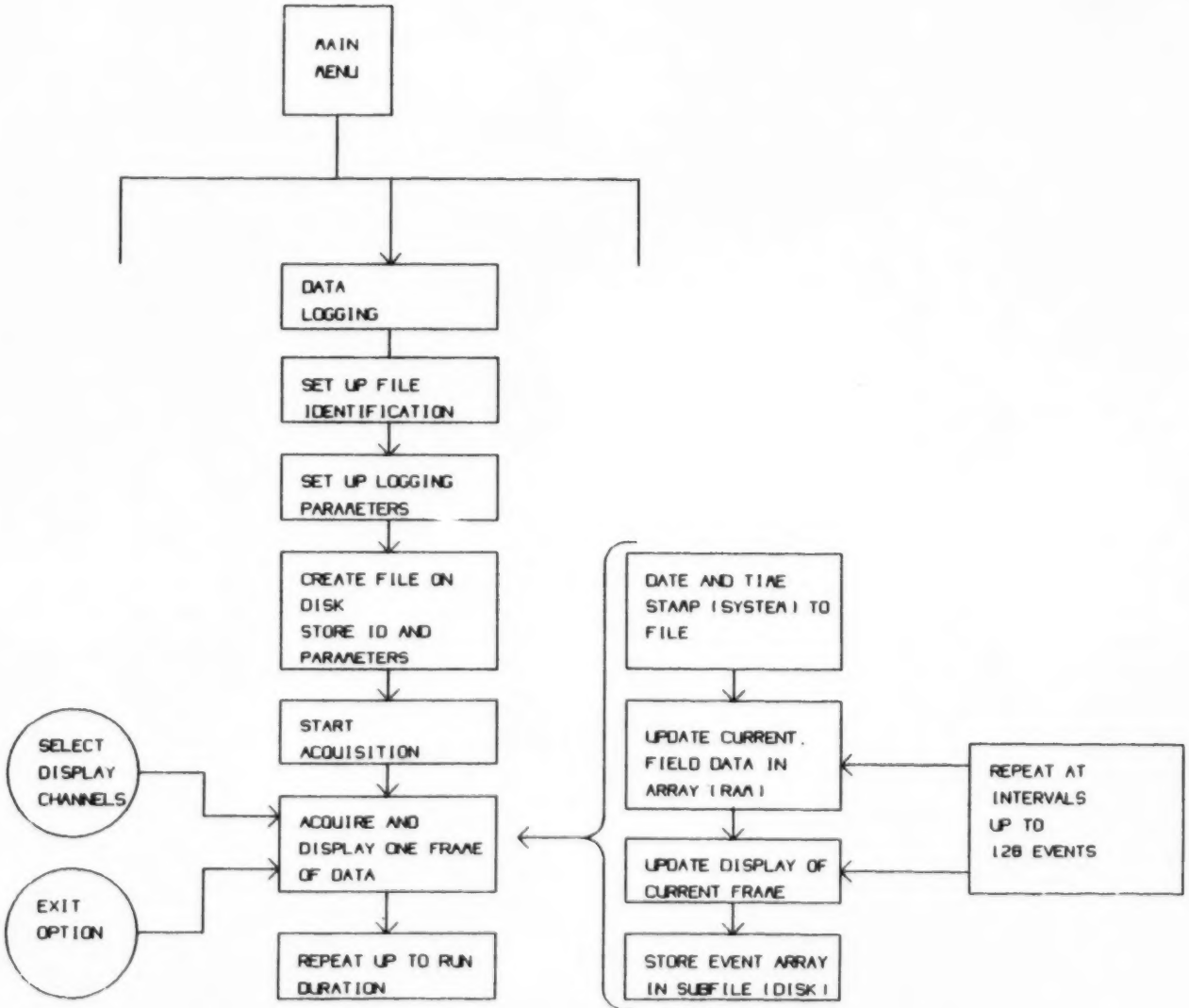

MAIN MENU
DATA LOGGING
SET UP FILE IDENTIFICATION
SET UP LOGGING PARAMETERS
CREATE FILE ON DISK STORE ID AND PARAMETERS
START ACQUISITION
ACQUIRE AND DISPLAY ONE FRAME OF DATA
REPEAT UP TO RUN DURATION
SELECT DISPLAY CHANNELS
EXIT OPTION
DATE AND TIME STAMP (SYSTEM) TO FILE
UPDATE CURRENT FIELD DATA IN ARRAY (RAM)
UPDATE DISPLAY OF CURRENT FRAME
STORE EVENT ARRAY IN SUBFILE (DISK)
REPEAT AT INTERVALS UP TO 128 EVENTS

FIGURE A2

DATA LOGGINE MENU

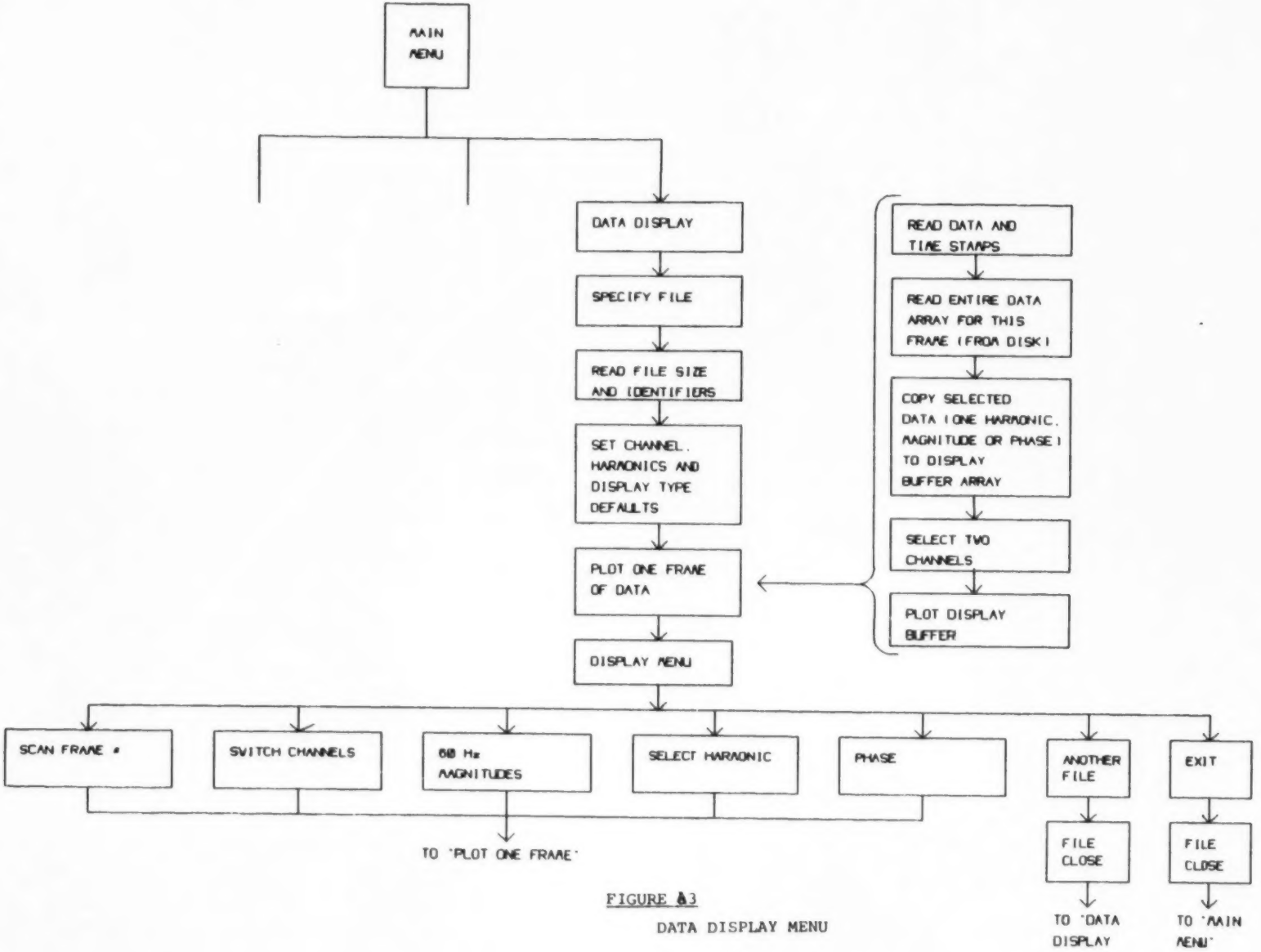


FIGURE A3

DATA DISPLAY MENU

```
1: NAME FOR FILE      :     D:\FIELDDATA\RES2

2: IDENTIFYING NAME   :     J. Q. PUBLIC

3: LOCATION           :     101 Anystreet

4: SEQUENCE #         :     001

5: ADDITIONAL REMARKS :
3-axis coil in family room.
Auxiliary coil on patio.


CHANGE ANY INPUT DATA ? INPUT ITEM # OR RETURN _
```

```
INPUT SAMPLE INTERVAL (MINUTES)
      DEFAULT 0.25 MINUTES    2.0
INPUT RUN DURATION (HOURS)
      DEFAULT 24 HOURS   18.0

SAMPLE INTERVAL =  2.0000 MINUTES
RUN DURATION    =  18.0000 HOURS

ACCEPT THESE VALUES ? Y/N _
```

FIGURE A4

FILE CREATION DIALOGUE

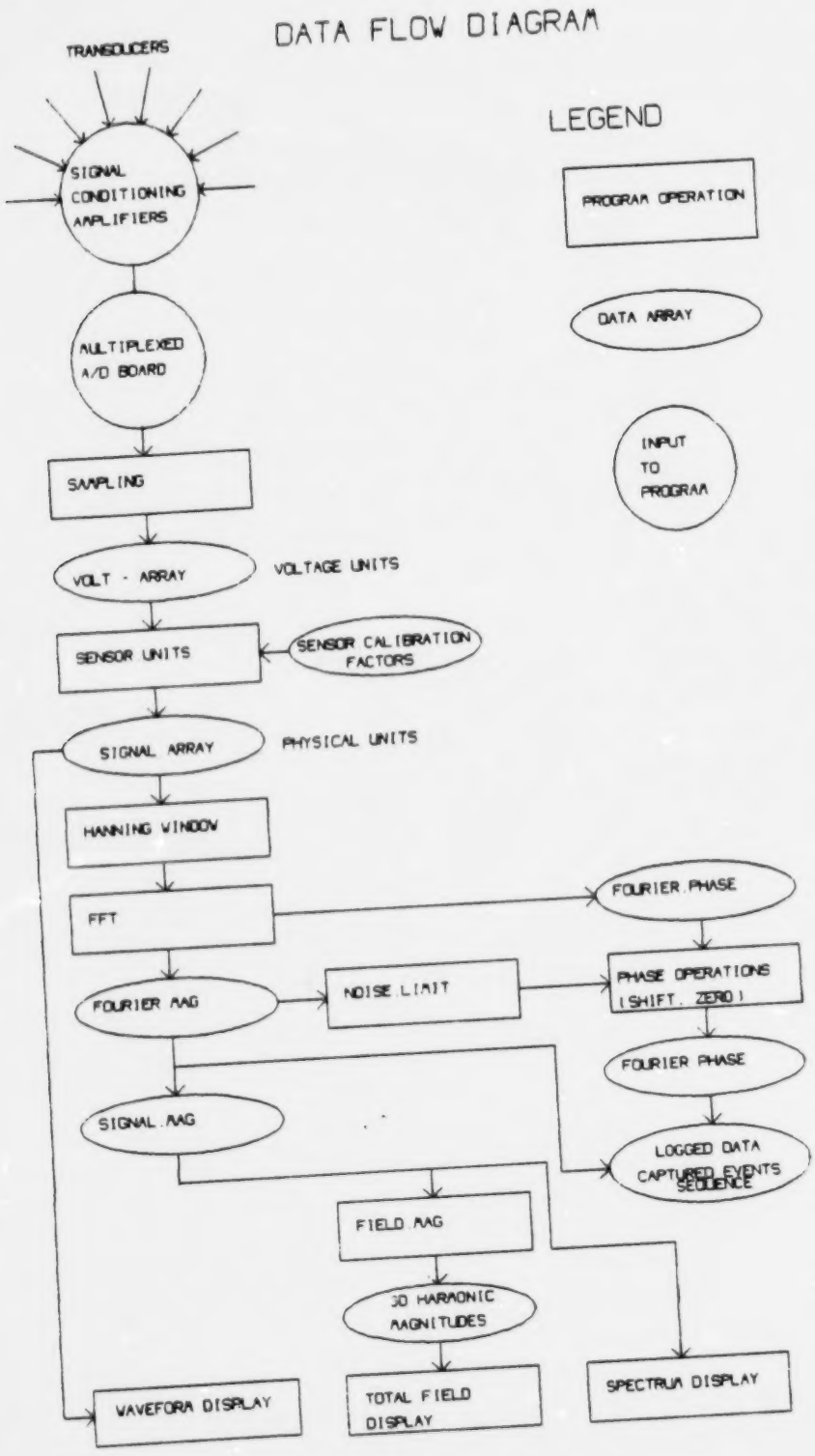

DATA FLOW DIAGRAM
TRANSDUCERS
SIGNAL CONDITIONING AMPLIFIERS
MULTIPLEXED A/D BOARD
SAMPLING
VOLT - ARRAY
VOLTAGE UNITS
SENSOR UNITS
SENSOR CALIBRATION FACTORS
SIGNAL ARRAY
PHYSICAL UNITS
HANNING WINDOW
FFT
FOURIER PHASE
FOURIER MAG
NOISE LIMIT
PHASE OPERATIONS (SHIFT, ZERO)
FOURIER PHASE
SIGNAL MAG
LOGGED DATA CAPTURED EVENTS SEQUENCE
FIELD MAG
3D HARMONIC MAGNITUDES
WAVEFORM DISPLAY
TOTAL FIELD DISPLAY
SPECTRUM DISPLAY
LEGEND
PROGRAM OPERATION
DATA ARRAY
INPUT TO PROGRAM

FIGURE A5

```
: SAMPLE

    INIT
    DT2825.ACQUIRE


    CONVERT                 \ digital to voltage conversion
    SENSOR.UNITS            \ voltage to physical units
    HANNING.WINDOW          \ create window function
    HARMONICS               \ fourier analysis of sampled signals
    RELATIVE.PHASE          \ refer phases to voltage fundamental
    PHASE.SHIFT             \ adjust phases for sample intervals
    PHASE.ZERO              \ ignore phases of harmonics < specified
                            \       amplitude
    PHASE.RANGE             \ return phases to range (-180,180) deg
    PHYSICAL.UNITS          \ calibration in frequency if required
    FIELD.MAGNITUDE         \ 3-dimensional field magnitude

;
```

FIGURE A6

WAVEFORM SAMPLING AND PROCESSING

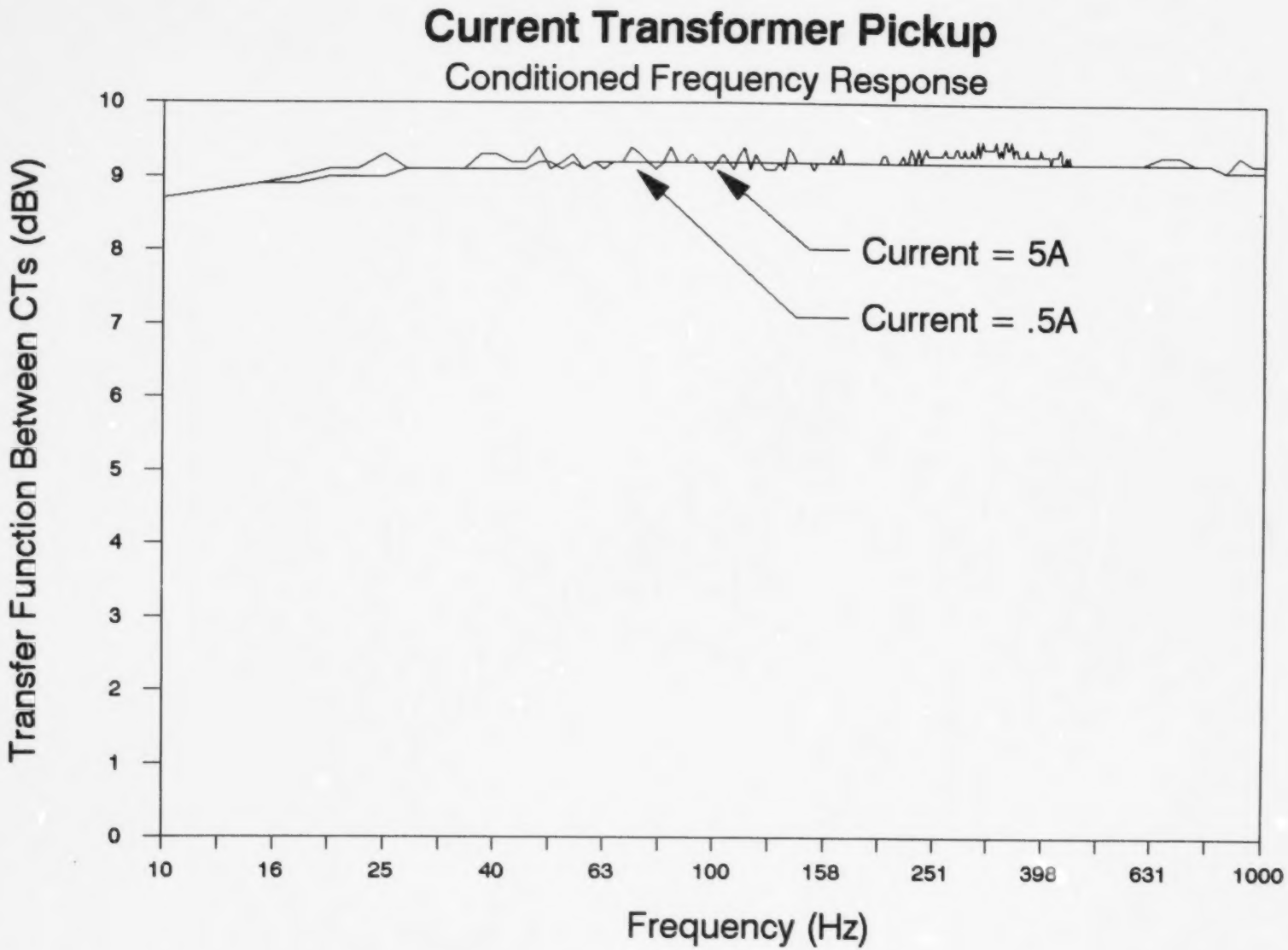


FIGURE A7
Magnitude Response in Current Channel

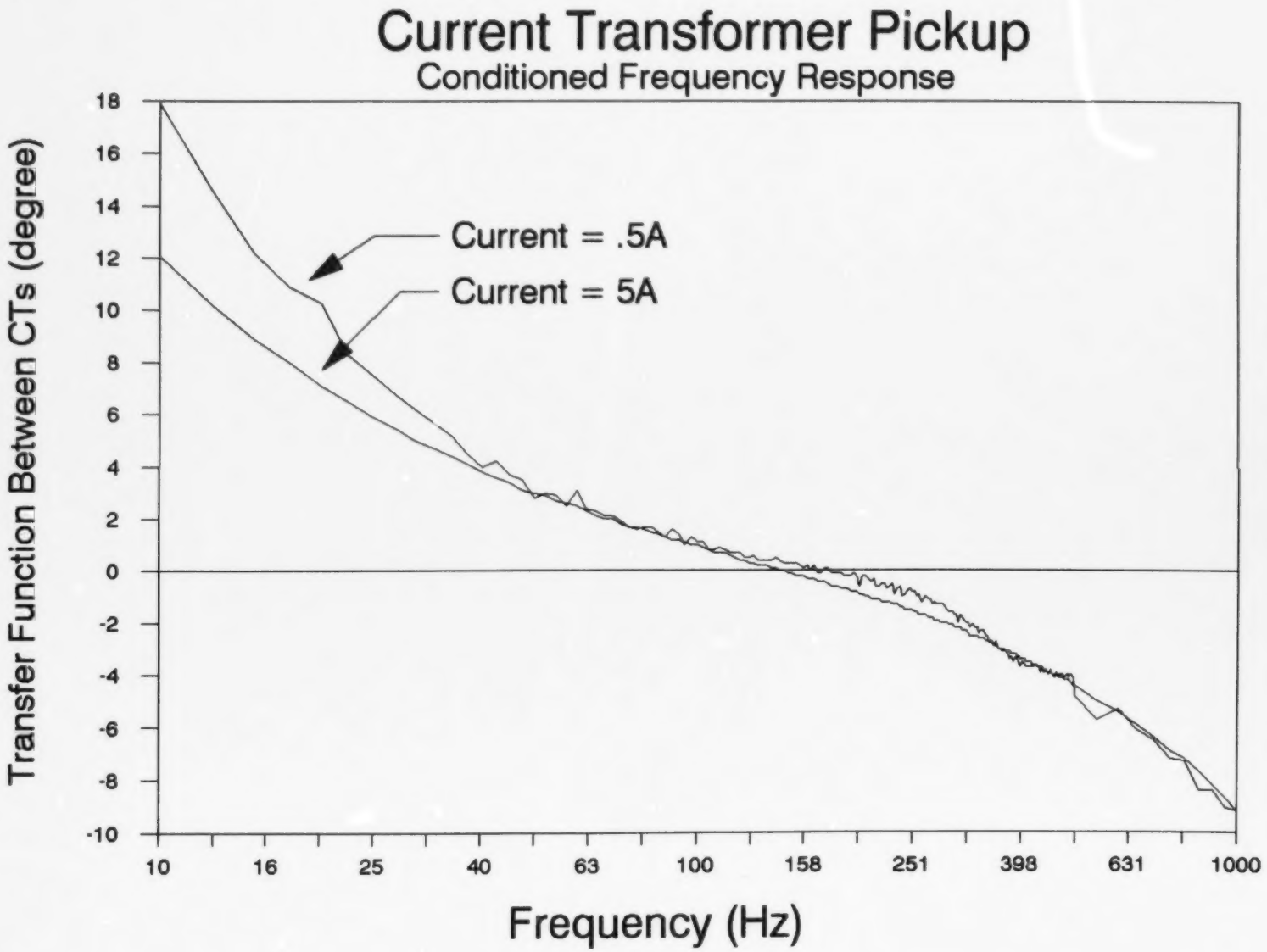


FIGURE A8
Phase Response in Current Channel

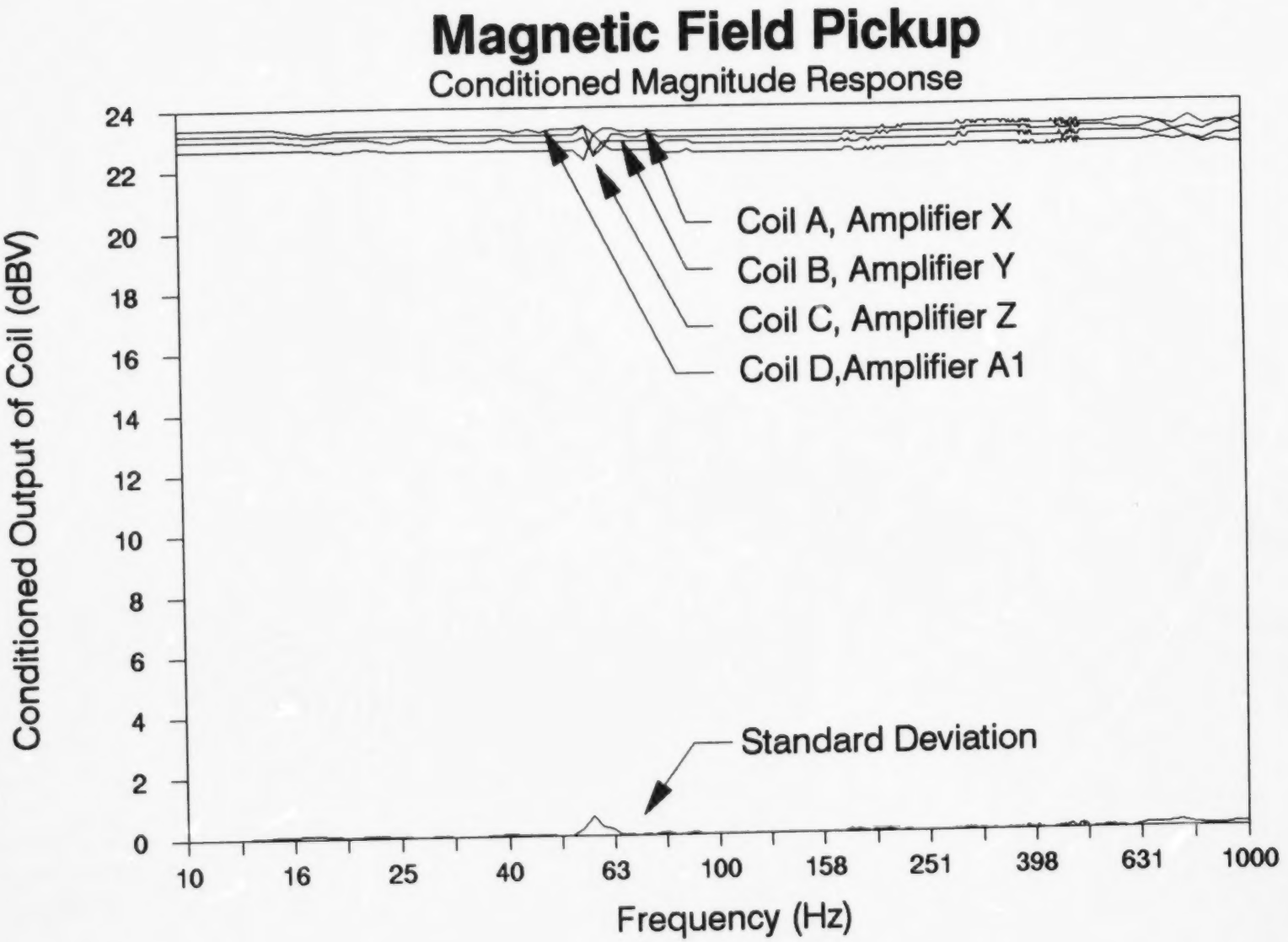


FIGURE A9
Magnitude Response in Field Channel

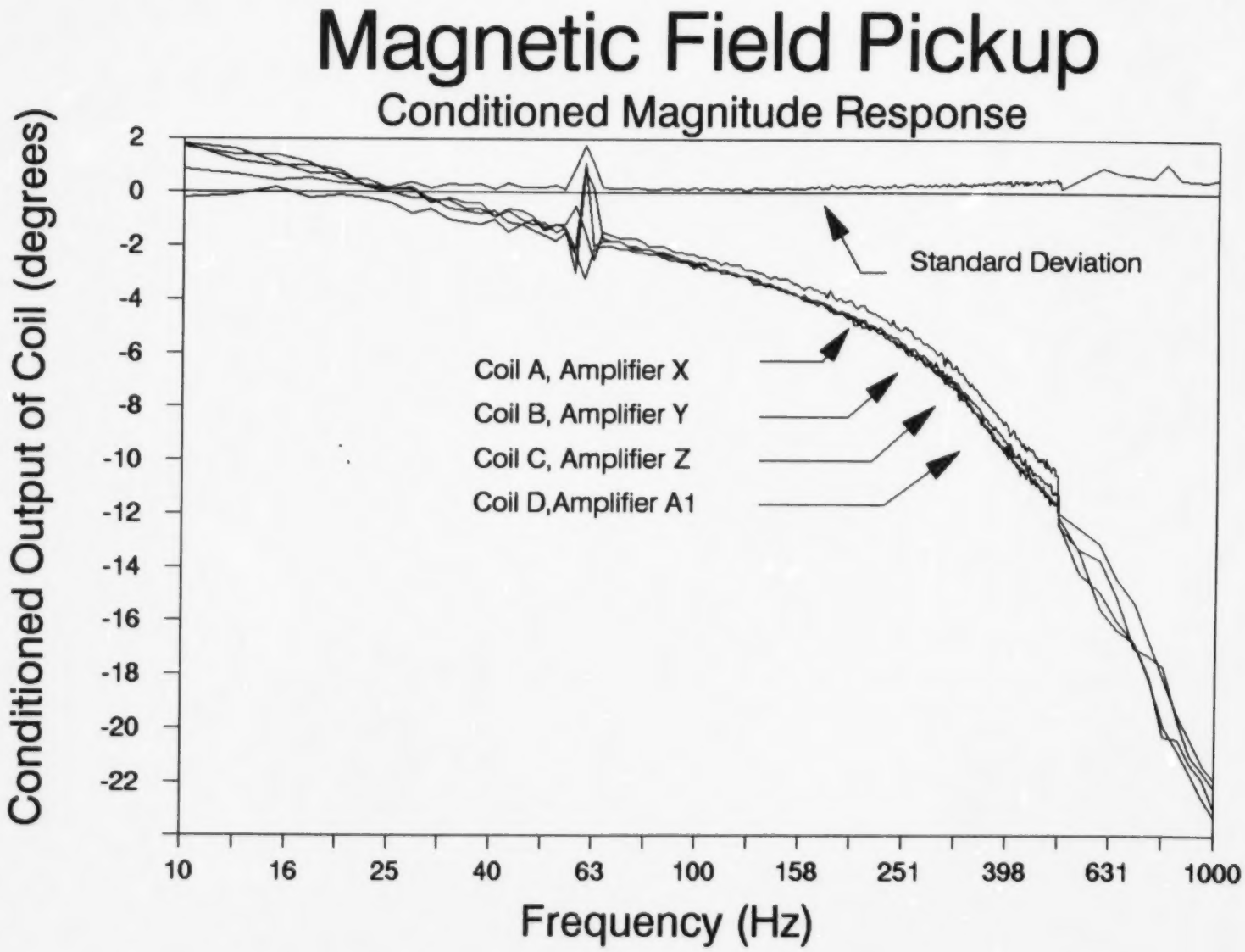
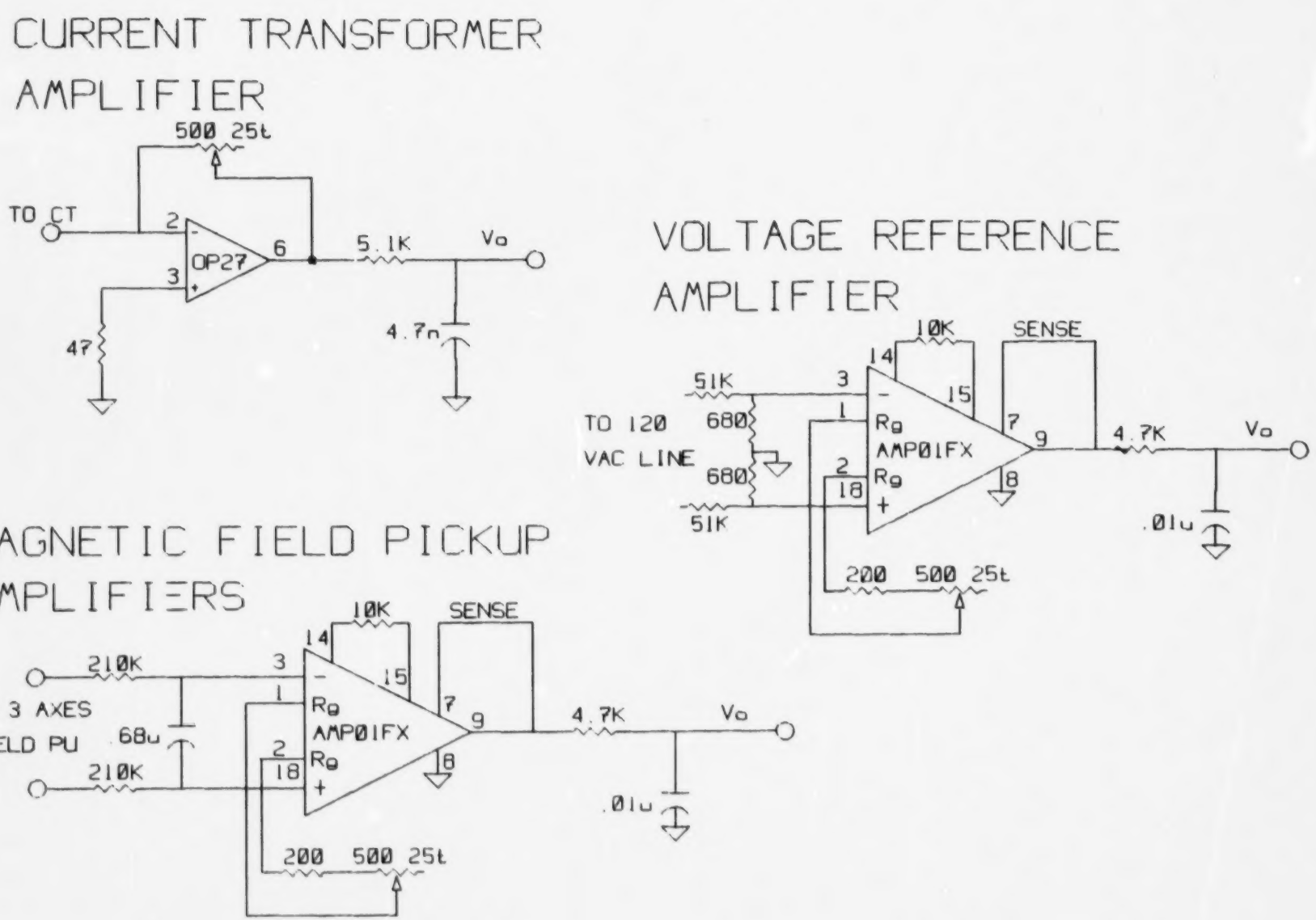


FIGURE A10
Phase Response in Field Channel

FIGURE A11


CURRENT TRANSFORMER AMPLIFIER
500 25t
TO CT
2
3
OP27
6
5.1K
Vo
47
4.7n
VOLTAGE REFERENCE AMPLIFIER
10K
SENSE
14
51K
3
15
TO 120 VAC LINE
680
1
RG
AMP01FX
7
9
4.7K
Vo
680
2
18
RG
+
8
51K
.01u
200 500 25t
MAGNETIC FIELD PICKUP AMPLIFIERS
10K
SENSE
14
210K
3
15
TO 3 AXES FIELD PU
1
RG
AMP01FX
7
9
4.7K
Vo
68u
2
18
RG
+
8
210K
.01u
200 500 25t